SDLA-312

ISA Security Compliance Institute —
Security Development Lifecycle Assurance -
Security Development Lifecycle Assessment

Version 6.3

December 2022

A. DISCLAIMER

ASCI and all related entities, including the International Society of Automation (collectively, “ASCI”) provide all materials, work products and, information (‘SPECIFICATION’) AS IS, WITHOUT WARRANTY AND WITH ALL FAULTS, and hereby disclaim all warranties and conditions, whether express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of reliability or availability, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence, all with regard to the SPECIFICATION, and the provision of or failure to provide support or other services, information, software, and related content through the SPECIFICATION or otherwise arising out of the use of the SPECIFICATION. Also, there is no warranty or condition of title, quiet enjoyment, quiet possession, correspondence to description, or non-infringement with regard to the SPECIFICATION.

Without limiting the foregoing, ASCI disclaims all liability for harm to persons or property, and users of this SPECIFICATION assume all risks of such harm.

In issuing and making the SPECIFICATION available, ASCI is not undertaking to render professional or other services for or on behalf of any person or entity, nor is ASCI undertaking to perform any duty owed by any person or entity to someone else. Anyone using this SPECIFICATION should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

B. EXCLUSION OF INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES

To the maximum extent permitted by applicable law, in no event shall ASCI or its suppliers be liable for any special, incidental, punitive, indirect, or consequential damages whatsoever (including, but not limited to, damages for loss of profits or confidential or other information, for business interruption, for personal injury, for loss of privacy, for failure to meet any duty including of good faith or of reasonable care, for negligence, and for any other pecuniary or other loss whatsoever) arising out of or in any way related to the use of or inability to use the SPECIFICATION, the provision of or failure to provide support or other services, information, software, and related content through the SPECIFICATION or otherwise arising out of the use of the SPECIFICATION, or otherwise under or in connection with any provision of this SPECIFICATION, even in the event of the fault, tort (including negligence), misrepresentation, strict liability, breach of contract of ASCI or any supplier, and even if ASCI or any supplier has been advised of the possibility of such damages.

C. OTHER TERMS OF USE

Except as expressly authorized by prior written consent from the Automation Standards Compliance Institute, no material from this document owned, licensed, or controlled by the Automation Standards Compliance Institute may be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, except for non-commercial use only, provided that you keep intact all copyright and other proprietary notices. Modification of the materials or use of the materials for any other purpose, such as creating derivative works for commercial use, is a violation of the Automation Standards Compliance Institute’s copyright and other proprietary rights.

version	date	changes
4.52	2018.01.31	Initial version aligned with IEC 62443-4-1
5.5	2019.08.04	Add SDLA-SVV-3A3, SDLA-SVV-3A4, SDLA-SVV-3, Appendix B; modify SDLA-SR-2i, SDLA-SI-1C-1, SDLA-SVV-3A2, SDLA-SVV-3A5 (previously 3A3), SDLA-SUM-2
5.7	2020.06.19	Modify SM-12, SUM-2, SUM-3, SUM-5; throughout, require documented process for all organization validation activities
6.3	2022.12.07	Incorporate errata from SDLA-102 v3.11; change "patch" to "update" in SUM-2 and SUM-3

Practice 1	Security Management (SM)	The purpose of the security management practice is to ensure that the security-related activities are adequately planned, documented and executed throughout the product's lifecycle
Practice 2	Specification of Security Requirements (SR)	The processes specified by this practice are used to document the security capabilities that are required for a product along with the expected product security context
Practice 3	Secure by Design (SD)	The processes specified by this practice are used to ensure that the product is secure by design including defence in depth
Practice 4	Secure Implementation (SI)	The processes specified by this practice are used to ensure that the product features are implemented securely
Practice 5	Security Verification and Validation Testing (SVV)	The processes specified by this practice are used to document the security testing required to ensure that all of the security requirements have been met for the product and that the security of the product is maintained when it is used in its product security context
Practice 6	Security Defect Management (DM)	The processes specified by this practice are used for handling security-related issues of a product that has been configured to employ its defence in depth strategy (Practice 3) within the product security context (Practice 2)
Practice 7	Security Update Management (SUM)	The processes specified by this practice are used to ensure security updates associated with the product are tested for regressions and made available to product users in a timely manner
Practice 8	Security Guidelines (SG)	The processes specified by this practice are used to provide documentation that describes how to integrate, configure, and maintain the defence in depth strategy of the product in accordance with its product security context

Normative references - The following pair of references provide the same technical standard, as published by the organizations ANSI/ISA and IEC.

ANSI/ISA-62443-4-1-2018 Security for industrial automation and control systems Part 4-1: Secure product development lifecycle requirements
IEC 62443-4-1:2018 Security for industrial automation and control systems Part 4-1: Secure product development lifecycle requirements

Informative references

[CSA-300] ISCI Component Security Assurance – ISASecure certification requirements, as specified at https://www.ISASecure.org
[SSA-300] ISCI System Security Assurance – ISASecure certification requirements, as specified at https://www.ISASecure.org

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SM-1	Development Process	A general product development/maintenance/support process shall be documented and enforced that is consistent and integrated with commonly accepted product development processes (for example, ISO 9001 [13] certified processes) that include but are not limited to: a) configuration management with change permission controls and audit record logging, b) product description and requirements definition with requirements traceability, c) software or hardware design and implementation practices, such as modular design; d) repeatable testing verification and validation process; e) review and approval of all development process records; and f) life-cycle support.	SDLA-SM-1A	Perform the component or system validation activities from Appendix A (taken from SDLA v1) or verify that the assessment used in the development organization validation activity is current, applicable, and was applied to the product or system being evaluated.	Verify that the documented development process is compliant with the configuration management requirements in Appendix A (Taken from SDLA v1) by validating those requirements, or verify that the process has been assessed to be compliant with another standard that includes configuration management such as IEC 61508, CMMI, or ISO 90003.	SDLA-SMP-5, SDLA-SMP-6, SDLA-SMP-7	
X	X				SDLA-SM-1B-1	None.	Verify that the documented development process states that requirements must be documented for each product and that there is a process to review and approve changes to requirements.	None	
X	X				SDLA-SM-1B-2	For security requirements, verify that the types of traceability described in the process are actually done for the component or system being evaluated.	Verify that the documented development process states that requirements traceability is required and the type of traceability that is required is documented (e.g. Forward Traceability between requirements and validation test, Backward Traceability between requirements and validation test, Forward Traceability between Requirements and Architectural Design.		
X	X				SDLA-SM-1C	Verify that the component or system being evaluated has a documented software and hardware (if applicable) design.	Verify that the documented development process includes software and hardware (if applicable) design practices. Verify that these practices include items that promote modular design.	SDLA-DSD-1	
X	X				SDLA-SM-1D	Verify that the verification and validation tests specified by the development process were carried out on the component or system being evaluated.	Verify that the documented process includes verification and validation tests. The validation tests should provide coverage on all of the product requirements. The verification tests should include some level of module testing and integration testing.	None	
X	X				SDLA-SM-1E	Verify that the reviews and approvals of artifacts described in the development process were done for the latest major release.	Verify that the documented process includes steps to review and approve development process artifacts such as requirements specifications, design specifications, and test plans.	None	
X	X				SDLA-SM-1F	None.	None	None	Note that lifecycle support is really covered by all of the other requirements in IEC 62443-4-1 since they cover the different phases of the lifecycle. Therefore, there are no additional requirements for this item.
X	X	SM-2	Identification of Responsibilities	A process shall be employed that identifies the organizational roles and personnel responsible for each of the processes required by this standard.	SDLA-SM-2	Verify that all security related activities and that those responsible for carrying out the activities are listed in the project documentation.	Verify the documented standard development lifecycle requires that all security related activities and those responsible for carrying out the activities are documented.	SDLA-SMP-1.1	
X	X	SM-3	Identification of applicability	A process shall be employed for identifying products (or parts of products) to which this standard applies.	SDLA-SM-3	Verify that the system or product under evaluation is one where it has been determined and documented that the security development lifecycle applies to the entire product (not just a part).	Verify that a documented process for identifying which products (or parts of products) the security development lifecycle applies to, exists. Do some sample auditing to confirm that the process is being used on the products identified by this process. At least 3 products should be reviewed in the sample auditing, unless there are not that many products identified by this process. In that case all products identified by this process should be reviewed.	None	
X	X	SM-4	Security expertise	A process shall be employed for identifying and providing security training and assessment programs to ensure that personnel assigned to the organizational roles and duties specified in 5.3, SM-2 – Identification of responsibilities, have demonstrated security expertise appropriate for those processes.	SDLA-SM-4	Verify that there is evidence of the competence of all people assigned processes defined in SDLA-SM-2 for the component or system being evaluated. This evidence can take the form of experience and qualifications, performance reviews, tests, or other assessments. Verify that everyone involved in software development has received the appropriate training and that this training and associated testing / demonstration of baseline competency has been documented.	Verify that company has a documented procedure to assess that personnel assigned to processes defined in SDLA-SM-2 have demonstrated security expertise appropriate for those processes. Verify that the documented development process states that for each defined role a list of required security training must be created and tracking who attends that training must be done. Verify that the required security training has been identified and that at least some developers have been trained.	SDLA-SMP-1.4, SDLA-SMP-1.5	Engineers must understand what it takes to build and deliver secure features; not how to develop security features. These skills are currently not taught in most colleges and universities and on average most software engineers know very little about software security.

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SM-5	Process scoping	A process, that includes justification by documented security analysis, shall be employed to identify the parts of this standard that are applicable to a selected product development project. Justification for scoping the level of compliance of a project to this standard shall be subject to review and approval by personnel with the appropriate security expertise (see SM-4).	SDLA-SM-5	If tailoring was done for the development of the component or system under evaluation, verify that a documented security analysis was done. Verify that any items tailored out were done so for a valid security reason (not for cost, scheduling or other purely business purposes). See Development Organization Validation activity column for examples of acceptable and unacceptable reasons. If the assessor is uncertain of the validity of a security reason, ISCI may be consulted for an opinion (without revealing customer name).	If company does not have a tailoring process, and they just apply all parts of the standard all of the time, then this requirement is met. However, if they do have a tailoring process defined in their documented process, verify that the tailoring must be justified by a documented security analysis. Review a project that tailoring was done (if one exists) and verify that a documented security analysis was done. The security analysis should include the reasons why an item has been tailored out, and should justify why not including this step will not have an adverse affect on security. The assessor should determine if the justification is reasonable based on his knowledge and experience. Below are some examples of reasonable and non-reasonable arguments: Reasonable: The product does not contain software, therefore a security coding standard is not needed. Reasonable: No communication interfaces or parsers were changed in this release, therefore, fuzz testing, which was run on the previous release, does not need to be repeated. Unreasonable: The product is very simple and therefore no threat model will be created. Unreasonable: The schedule is very tight, so no penetration testing will be done.	None	
X	X	SM-6	File Integrity	A process shall be employed to provide an integrity verification mechanism for all scripts, executables and other important files included in a product.	SDLA-SM-6	Verify that a method was used to assure users that the code/files did actually come from the supplier and to verify that that they have not been tampered with. If a method other than digital signing was used, verify that the method meets the intent of this requirement.	Verify that the documented development process states that a method must be used to assure users that the code, scripts and other important files did actually come from the supplier and to verify that the files have not been tampered with since their publication.	None	
X	X	SM-7	Development environment security	A process that includes procedural and technical controls shall be employed for protecting the product during development, production and delivery. This includes protecting the product or product update (patch) during design, implementation, testing and release.	SDLA-SM-7	None.	Verify that there are documented procedural and technical controls in place and that they cover the development environment, production, and delivery. Verify the procedures specifically include methods (both procedural and technical) to protect private keys. Controls for private keys should be based on recommended practices from a well known industry source (for example see Key Management best practices from OWASP) or include the following at a minimum: 1. Keys should never be stored in plaintext format. 2. Ensure all keys are stored in a hardware storage device such as a hardware security module (HSM), smart card, or USB token. 3. Ensure that keys and cryptographic operation is done inside an area that has limited physical access. 4. The number of people with access to the keys should be limited to those users who require access. Pick a development project and sample some of these methods to determine if they are being followed for that project.	SDLA-SMP-4, SDLA-SMP-4.1	
X	X	SM-8	Controls for private keys	The supplier shall have procedural and technical controls in place to protect private keys used for code signing from unauthorized access or modification.	SDLA-SM-8	Determine if there are private keys used in the component or system under evaluation. If so, review how those keys are stored and protected. Verify that there are both procedural and technical controls in place to protect them and verify that they are being followed.			

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SM-9	Security requirements for externally provided components	A process shall be employed to identify and manage the security risks of all externally provided components used within the product.	SDLA-SM-9	Verify that externally provided components were identified and documented for the component or system being evaluated. Verify that for each such component, the security risks were identified and documented and that a method for managing or mitigating each of those risks was documented. Pick a few of those risks and verify that the method for managing or mitigating those risks was carried out and was appropriate for the risk.	Verify that there is a documented process in place to identify any externally provided components used in each product. Verify that there is a documented process in place to identify and manage the security risks of all such components for the life of the product. Verify that the security risks of all such components are required by the documented process to be re-evaluated periodically as security risks change over time. Pick a product and verify that externally provided components were identified and documented. Verify that for each such component, the security risks were identified and documented and that a method for managing or mitigating each of those risks was documented.	None	
X	X	SM-10	Custom developed components from third-party suppliers	A process shall be employed to ensure that product development life-cycle processes for components from a third-party supplier conform to the requirements used in this document when they meet the following criteria: a) the components are developed specifically for a single supplier for a specific purpose; and b) the components can have an impact on security.	SDLA-SM-10	Determine if there are any third-party components developed specifically for the supplier included in the component or system under evaluation. If so, ensure that that the SDL processes required by the suppliers development procedures were applied to those components or sufficient evidence has been documented to indicate that such components have no impact on security.	Verify that there is a documented procedure indicating that all third party components developed specifically for this supplier are subject to the same security development lifecycle for the life of the product unless those components can be shown to have no impact on security. In order to show that a component has no impact on security, the supplier should have a process in places that determines whether a component impacts security or not. See comments for an example of a way to determine if a component has an impact on security.	None	The following types of changes are among those that usually have an impact on security: -Code listening on the network or connecting to the network -Code with prior vulnerabilities identified -Code executing with high privilege (for example SYSTEM, administrator, root) -Security related code (for example, authentication, authorization, cryptographic and firewall code) -Code that parses data structures from potentially untrusted sources -Setup code that sets access controls or handles encryption keys or passwords
X	X	SM-11	Assessing and Addressing security-related issues	A process shall be employed for verifying that a product or a patch is not released until its security-related issues have been addressed and tracked to closure (See 10.5, DM-4:Addressing security-related issues). This includes issues associated with a) Requirements (see Clause 6, Practice 2 - Specification of Security requirements); b) secure by design (see Clause 7, Practice 3 - Secure by design); c) implementation (see Clause 8, Practice 4 - Secure implementation); d) verification/validation (see Clause 9, Practice 5 - Security verification and validation testing); and e) defect management (see Clause 10, Practice 6 - Management of security-related Issues).	SDLA-SM-11	For the product or system being evaluated, randomly review artifacts from development such as meeting minutes, test results and threat models and identify issues and verify whether they were documented and tracked to closure.	Verify that a documented procedure exists to document and track security-related issues to closure. Verify that this procedure includes issues found in all practices listed in the requirement.	SDLA-SMP-2 SDLA-SPV-1.9	
X	X	SM-12	Process verification	A process shall be employed for verifying that, prior to product release, all applicable security-related processes required by this specification (See SM-5: Process Scoping) have been completed with records documenting the completion of each process.	SDLA-SM-12	Verify that the verification process defined in this requirement was carried out for the system or component being evaluated prior to last product release.	Verify that there is a documented process for verifying that, prior to product release, all security-related processes required by this specification have been carried out. Verify that this requirement applies to all types of releases (initial release, major release, minor release, security patches or hot fixes). Note that the process scoping requirement (SM-5) applies here as well. So for a given release, if items have been deemed to be out of scope as per SM-5, then no verification of those items is needed. Verify that a method of enforcing the process exists in documented organization policy. Verify that individuals responsible for enforcement of policy and auditing of security-related processes are aware of their responsibilities. Pick a product and verify that there is evidence that execution of security-related processes was audited and the method of enforcement was carried out.	None	

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SM-13	Continuous Improvement	A process shall be employed for continuously improving the SDL. This process shall include the analysis of security defects in component/subsystem/system technologies that escape to the field.	SDLA-SM-13	None.	Verify that there is a documented process in place to review any security defects that reach the field and apply lessons learned to improve the development process. Verify that there is a documented process in place to periodically review how the development process can be improved based on field issues, changes to the security landscape, and experience.	SDLA-SRP-2.7	

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SR-1	Product security context	A process shall be employed to ensure that the intended product security context is documented.	SDLA-SR-1	Verify Security Requirements Specification (SecRS) includes a description of the operating environment Verify SecRS identifies and explains assumptions about the intended usage of the product and the environment	Verify SecRS includes a description of the operating environment for any product developed according to the process currently being evaluated. Or verify that the development process or SecRS template states that the SecRS must include a statement of expected security environment. May verify SecRS for any component or system developed according to the process being evaluated identifies and explains assumptions about the intended usage of the product and the environment. Or may verify that the development process or SecRS template states that assumptions about intended usage of the product and the environment are included in the SecRS.	SDLA-SRS-3, SDLA-SRS-3.1	
X	X	SR-2	Threat model	A process shall be employed to ensure that all products shall have a threat model specific to the current development scope of the product with the following characteristics (where applicable):	SDLA-SR-2	Verify that the threat model is up to date based on the most recent design changes.	Verify that there is a documented policy that the threat model should be updated when the design changes.	SDLA-SRA-3.2 SDLA-SRA-3	
X	X	SR-2	Threat Model	A process shall be employed to ensure that all products shall have a threat model specific to the current development scope of the product with the following characteristics (where applicable): a) correct flow of categorized information throughout the system; b) trust boundaries; c) processes; d) data stores; e) interacting external entities;	SDLA-SR-2A	Verify that data flow diagrams are included in the threat model. The DFD should include a context diagram and detailed lower level data flows. If another method of modeling system behavior is included, verify that it documents data flows. Verify that the data flow diagram includes processes, data stores trust boundaries and interacting external entities.	Verify that the documented development process requires that data flow diagrams or equivalent method are included in the threat model. If an equivalent method is used verify that that method includes the documentation of dataflow (For example UML sequence diagrams). Or verify that the threat model for any component or system developed according to the same process being evaluated includes data flow diagrams or an equivalent method.	SDLA-SRA-3.7 SDLA-SRA-3.3 SDLA-SRA-3.8	
X	X	SR-2	Threat model	A process shall be employed to ensure that all products shall have a threat model specific to the current development scope of the product with the following characteristics (where applicable): f) internal and external communication protocols implemented in the product g) externally accessible physical ports including debug ports h) circuit board connections such as Joint Test Action Group (JTAG) connections or debug headers which might be used to attack the hardware	SDLA-SR-2F	Verify that the threat model for the product being evaluated includes the following: f) internal and external communication protocols implemented in the product g) externally accessible physical ports including debug ports (unless there are no such ports included in the product) h) circuit board connections such as Joint Test Action Group (JTAG) connections or debug headers which might be used to attack the hardware (unless there are no such connections on the product). For systems, verify that the threat model includes all accessible points of entry submitted by the supplier as required by SSA-300.	Not Required		
X	X	SR-2	Threat model	A process shall be employed to ensure that all products shall have a threat model specific to the current development scope of the product with the following characteristics (where applicable): i) potential attack vectors including attacks on the hardware if applicable j) potential threats and their severity as defined by a vulnerability scoring system (for example, CVSS) l) security-related issues identified	SDLA-SR-2i	Verify that threat model documents a list of threats identified in the threat modeling process. For a system, the two specific threats examined below do not comprise a comprehensive list of threats, but are included in such a list. For a system, verify that the list includes the potential to exploit any capabilities of each system component not intended to be used when functioning as a component of the system (or related error feedback). These could include native component identification, authentication, cryptographic, logging or backup functions are unused and possibly superseded by system level capabilities. These may also include interfaces (human, machine to machine, wired or wireless) or input/output of specific data not required by the system. For a system, verify that the list includes the potential for unintended modifications to software or firmware for any system component, prior to initialization or at runtime. (Example mitigations under SDLA-SR-2K below may be a secure boot capability at initialization and runtime automated or manual verification of hashes for executables.)	Verify that the documented development process requires that a list of threats are included in the threat model. Verify that the threat model for any component or system developed according to the same process being evaluated includes a list of threats.	SDLA-SRA-3.9	A security related issue is characteristic of the design or implementation of the product that can potentially affect the security of the product. Each threat in the model is a security related issue.

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SR-2	Threat model	A process shall be employed to ensure that all products shall have a threat model specific to the current development scope of the product with the following characteristics (where applicable): j) potential threats and their severity as defined by a vulnerability scoring system (for example, CVSS)	SDLA-SR-2J	Verify that if the threat model identifies any threats previously identified, that are no longer relevant to the product, that the threat model includes a documented rationale for why this is the case. Verify that each threat in the threat model that remains relevant is either assigned a risk or severity score in accordance with the documented process, or there is a documented rationale for not further mitigating the threat that includes a description of consequence and likelihood for this threat. Verify that the rationale evaluates these factors for an attacker with characteristics associated with the capability security level of the product. Verify that any threat that relies upon a published CVE report that includes a base score, is assigned either the published CVSS base score, or an environmental or temporal CVSS score that incorporates adjustments to the base score. Verify that the score or rationale associated with each threat in the threat model is consistent with the product security context.	Verify that the documented development process requires that each threat in the threat model that remains relevant to the product, is either assigned a risk or severity score, or a documented rationale is provided for not further mitigating that threat. Verify that the scoring process either uses an accepted scoring methodology, or the method is well-documented and rationalized, takes into account likelihood and consequence of the threat, and clearly defines how scores are assigned. Verify that the documented development process requires that either the CVSS base score, or an environmental or temporal CVSS score that incorporates adjustments to the base score, is assigned to threats that rely upon a published CVE report that includes a base score. Verify that the documented process states that where a documented rationale is provided for not further mitigating a threat, it includes a description of consequence and likelihood, where these factors are evaluated for an attacker with characteristics associated with the capability security level of the product.	SDLA-SRA-3.10	At the threat modeling stage, CVSS scoring is not required for all threats. For threats not based on a published CVE report that includes a score, other accepted or well-rationalized scoring methods may be used, or a formal score may not be calculated if a threat is no longer relevant or a clear rationale for not further mitigating the threat is provided. (A score assigns a result from among a pre-defined set of possible results (which may be numerical); a rationale documents the reason for a decision.) A threat remains relevant to a product if any residual risk remains. A threat is considered relevant, even if it has very low probability and/or consequence for attackers with characteristics associated with the capability security level of the product. In a few cases, a threat previously judged relevant, may later become no longer relevant to a product. As examples, the threat may be been “designed out” (such as removal of a product interface), and/or the product implementation, or security context modified to render the threat no longer applicable for the product. The fact that an environmental CVSS score can be used allows adjustment to the base score to take into account known common factors that would increase or decrease risk due to the threat in customer environments, which may differ from related risk in IT environments. The fact that the temporal CVSS score can be used allows the base score to be adjusted based on whether the vulnerability is unknown or not, whether exploits are available and whether there is a patch or work around for the problem.
X	X	SR-2	Threat model	A process shall be employed to ensure that all products shall have a threat model specific to the current development scope of the product with the following characteristics (where applicable): k) mitigations and/or dispositions for each threat	SDLA-SR-2K	Verify that all threats that have been assigned a score above the defined risk or severity score have a documented mitigation by one or more of the following methods: 1) defence in depth strategy or design change 2) requiring compensating controls at the time of integration 3) addition of one or more security requirements and/or capabilities 4) disabling or removing features 5) creating a remediation plan to fix the problem	Verify that a documented procedure exists stating that all threats assigned a risk or severity score, threats above a defined risk or severity score must be mitigated. Verify that the defined score is well-specified and rationalized, and at a minimum includes all risks that are classified as critical or high when using a CVSS score.	SDLA-SRA-3.11	The defined risk or severity score can differ by capability security level of the product, but this is not required.
X	X	SR-2	Threat model	All products shall have an up-to-date threat model with the following characteristics: m) 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.	SDLA-SR-2M	Inspect the threat model and verify that external dependencies are listed or that it explicitly states that there are none.	Verify that the documented development process requires that external dependencies are included in the threat model. Or verify that the threat model for any component or system developed according to the same process being evaluated includes external dependencies.	SDLA-SRA-3.5	
X	X	SR-2	Threat model	The threat model shall be reviewed periodically (at least once a year) for released products and updated if required in response to the emergence of new threats to the product even if the design does not change	SDLA-SR-2N	Verify that the threat model has been updated within the past year.	Verify that a documented procedure exists stating that threat models should be updated periodically even if the design does not change. Verify that this period is at least once per year. In the case of an SDLA certification renewal, pick a project or two and verify that this has been happening.	N/A	
X	X	SR-2	Threat model	The threat model shall be reviewed and verified by the development team to ensure that it is correct and understood. Any issues identified in the threat model shall be addressed as defined in 10.4, DM-3 – Assessing security-related issues, and 10.5, DM-4 – Addressing security related issues.	SDLA-SR-2O	Verify that the threat model review was carried out, that minutes were documented for the meeting, and all action items have been dispositioned as defined in DM-4.	Verify that a documented procedure exists stating that the threat model must be subject to an internal review by the development team to make sure that it is correct and understood.	N/A	
X	X	SR-3	Product security requirements	A process shall be employed for ensuring that security requirements are documented for the product/feature under development including requirements for security capabilities related to installation, operation, maintenance, and decommissioning	SDLA-SR-3	Verify security requirements specification exists for component or system under evaluation and includes required security capabilities related to installation, operation, maintenance, and decommissioning if these phases are applicable. The specification can be in many forms such as a Microsoft Word document and may be part of another requirements specification.	Verify that the development process states that security requirements must be created and documented and include requirements for security capabilities related to installation, operation, maintenance and decommissioning. May verify that security requirements exist for any product developed under the process being certified.	SDLA-SRS-1	The SecRS doesn't need to be single document. Many organizations create a security requirements section in other requirements and customer documents.

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SR-4	Product security requirements content	A process shall be employed for ensuring that security requirements include the following information: a) the scope and boundaries of the component or system, in general terms in both a physical and a logical way; and b) the required capability security level (SL-C) of the product.	SDLA-SR-4	Verify the security requirements includes the scope and boundaries of the device in both a physical and logical way. Verify that the security requirements include the required capability security level of the component or system being evaluated.	May verify the security requirements include the scope and boundaries of the component or system in both a physical and logical way for any component or system developed under the process being certified.	SDLA-SRS-2.1	
X	X	SR-5	Security requirements review	A process shall be employed to ensure that security requirements are reviewed, updated as necessary and approved to ensure clarity, validity, alignment with the Threat Model (discussed in 6.3 SR-2 –Threat model), and their ability to be verified. Each of the following representative disciplines shall participate in this process. Personnel may be assigned to more than one discipline except for testers, who shall remain independent. a) Architects/developers (those who will implement the requirements); b) testers (those who will validate that the requirements have been met); c) customer advocate (such as sales, marketing, product management or customer support); and d) Security Advisor	SDLA-SR-5	Verify evidence that the requirements were reviewed for these specific qualities (e.g. details in meeting minutes or completion of review checklist) for the component or system being evaluated. Verify that at least one developer, tester, and customer advocate was involved in the review. Evidence of requirements review and approval on latest version of requirements (e.g. meeting minutes with version of requirements specification reviewed).	Verify that the development process or review checklist states that the requirements are analyzed for clarity, validity, and the ability to be verified. Verify that the development process states that at least one developer, tester, and customer advocate is involved in this review. Verify that the development process states that all changes to the requirements after the initial review are subject to an additional review using the same review criteria.	SDLA-SRS-9 and SDLA-SRS-10	

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements
X	X	SD-1	Secure design principles	A process shall be employed for developing and documenting a secure design that identifies and characterizes each interface of the product, including physical and logical interfaces, to include: a) an indication of whether the interface is externally accessible (by other products), or internally accessible (by other components of the product), or both; b) security implications of the product security context (see Clause 6, Practice 2 – Specification of security requirements) on the external interface; c) potential users of the interface and the assets that can be accessed through it (directly or indirectly); d) a determination of whether access to the interface crosses a trust boundary; e) security considerations, assumptions and/or constraints associated with the use of the interface within the product security context, including applicable threats; f) the security roles, privileges/rights and access control permissions needed to use the interface and to access the assets defined in c) above; g) the security capabilities and/or compensating mechanisms used to safeguard the interface and the assets defined in c) above, including input validation as well as output and error handling. h) the use of third-party products to implement the interface and their security capabilities; and i) documentation that describes how to use the interface if it is externally accessible. j) description of how the design mitigates the threats identified in the threat model	SDLA-SD-1	-Inspect the component or system architecture design description for the component or system being evaluated and verify that the design identifies and describes the exposed interfaces. Sample a few of the exposed interfaces defined in the design to confirm that items (a) through (j) from this requirement are documented for those interfaces. -Inspect the system architecture design and verify that the design shows how the system's devices and subsystems are connected, and how external actors are connected to the system. -Inspect the system architecture design and verify that the design shows all protocols used by all external actors to communicate with the system. -Inspect the component or system architecture design description and verify that trust boundaries are documented.	-Verify that the documented development process or software architecture design template indicates that a security design must be documented which identifies and characterizes each exposed interface of the component or system. Verify that there is either a checklist, a template, or a documented procedure which defines the information that must be documented for each interface, and that this matches items (a) through (j) from the requirement. -Verify that the documented development process or architecture design template indicates that trust boundaries must be documented as part of the architecture design. or, inspect the component or system architecture design description for any product developed with the process being evaluated and Verify that trust boundaries are documented.	SDLA-SAD-2.1 SDLA-SAD-4 SDLA-SAD-2 SDLA-DSD-1.1 SDLA-DSD-1.5
X	X	SD-2	Defence in depth design	A process shall be employed to implement multiple layers of defence using a risk based approach based on the threat model. This process shall be employed for assigning responsibilities to each layer of defence. NOTE 1 Each layer provides additional defence mechanisms NOTE 2 Each layer may be compromised; therefore, secure design principles are applied to each layer. NOTE 3 The objective is to reduce the attack surface of the subsequent layers	SDLA-SD-2	Examine the design for the component or system being evaluated. For a system design, verify that multiple layers of defence are included in the design and that each layers has clear responsibilities assigned. For a component design, verify that the design is not solely dependent on other components or layers for its security. Verify that a methodology to determine which layers of defence are required as defined for this project and followed.	Verify that the defence in depth concept is included as part of the documented design process or design guidelines. Verify that a design methodology to determine which layers of defence are required is included in the documented process or required on a per project basis.	

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements
X	X	SD-3	Security Design Review	A process shall be employed for conducting design reviews to identify, characterize, and track to closure security-related issues associated with each significant revision of the secure design including but not limited to: a) security requirements (Practice 2) that were not adequately addressed by the design, NOTE 1 Requirements allocation, including security requirements, is part of typical design processes. b) threats and their ability to exploit product interfaces, trust boundaries, and assets (SD-1 – Secure design principles), c) identification of design best practices (SD-4 – Secure design industry recommended practices) that were not followed (for example, failure to apply principle of least privilege) NOTE 2 Characterizing threats and their ability to exploit interfaces is often referred to as threat modeling.	SDLA-SD-3	Verify that security design reviews have been done for the product or system being evaluated. Look for evidence, such as a completed checklist, that the design review included checks on items (a) through (c) from the requirement. Examples of how checks on each of these items can be shown are as follows: (a) Traceability from security requirements to security design will demonstrate that requirements have been adequately addressed in the design (b) Traceability from threat mitigations to security design and security guidelines for users will demonstrate that threats have been addressed sufficiently. (c) A checklist of security best practices filled out in preparation for or during the design review will show that the review looked for sufficient best practices in the design. -Verify that issues identified in the design review have been documented in an issue tracking system where issues are tracked to closure.	Verify that the documented development process requires that security design reviews be performed on parts of the project that have been identified as relevant for security. Verify that security design reviews have been done for any product or system that has been developed according to the same process being evaluated. Verify that there is some sort of checklist or guideline which indicates items to check in the review and that the checklist includes items (a) through (c) from the requirement.	SDLA-SRA-1
X	X	SD-4	Secure design best practices	A process shall be employed to ensure that secure design best practices are documented and applied to the design process. These practices shall be periodically reviewed and updated. Secure design practices include but are not be limited to: a) least privilege (granting only the privileges to users/software necessary to perform intended operations); b) using proven secure components/designs where possible; c) economy of mechanism (striving for simple designs); d) using secure design patterns; f) all trust boundaries are documented as part of the design; and g) removing debug ports, headers and traces from circuit boards used during development from production hardware or documenting their presence and the need to protect them from unauthorized access.	SDLA-SD-4	Verify that some of the secure best practices defined in this requirement have been employed and documented in the development of the component or system being evaluated. Verify that the mechanism for ensuring that this requirement was followed was performed for the component or system being evaluated (e.g. a completed checklist can be found).	Verify that secure best practices are documented as part of the process, and that some mechanism is in place to ensure that they were followed (for example a review with a checklist). Verify that the process states that these best practices are periodically reviewed and updated. Verify that at a minimum the best practices include the items defined in (a) through (g) of this requirement. If this analysis is being applied to an SDLA renewal, verify that the security best practices have been updated since the initial certification.	SDLA-SAD-8 SDLA-DSD-2
X	X	SD-4	Secure design best practices	A process shall be employed to ensure that secure design best practices are documented and applied to the design process. These practices shall be periodically reviewed and updated. Secure design practices include but are not be limited to: e) attack surface reduction;	SDLA-SD-4E	Verify that work was done to reduce the attack surface, that this work was documented, and that any actions from this analysis have been completed.	Verify that the development process states that attack surface reduction techniques must be practiced and documented. Verify that documented evidence of attack surface reduction exists for any component or system developed using the same process being evaluated.	SDLA-SAD-6

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SI-1	Security implementation review	A process shall be employed to ensure that implementation reviews are performed for identifying, characterizing and tracking to closure security-related issues associated with the implementation of the secure design including: a) identification of security requirements (see Clause 6, Practice 2 – Specification of security requirements) that were not adequately addressed by the implementation; NOTE Requirements allocation, including security requirements, is part of typical design processes. b) identification of secure coding standards (see 8.4, SI-2 – Secure coding standards) that were not followed (for example, use of banned functions or failure to apply principle of least privilege); c) Static Code Analysis (SCA) for source code to determine security coding errors such as buffer overflows, null pointer dereferencing, etc. using the secure coding standard for the supported programming language. SCA shall be done using a tool if one is available for the language used. In addition, static code analysis shall be done on all source code changes including new source code. d) review of the implementation and its traceability to the security capabilities defined to support the security design (see Clause 7, Practice 3 – Secure by design); and e) examination of threats and their ability to exploit implementation interfaces, trust boundaries and assets (see 7.2, SD-1 – Secure design principles, and 7.3, SD-2 – defence in depth design).	SDLA-SI-1	Verify that some code and some hardware implementation has been reviewed, and that there is a clear list of what has been reviewed. Verify there is some evidence that the code review checklist was used during the review (such as a completed checklist or a statement about the checklist used in the code review). May verify that the code review results are documented along with the following information: name of the person who performed the code review, the date of the code review, the results of the code review and the name of the person responsible for fixing problems identified in the code review and a date or indication that all problems were fixed. Code review results can be documented electronically or via paper copies, but the results must be available to an auditor. Items identified in the code review that were not fixed should be identified along with an explanation as to why they were not fixed. The code review results should be inspected for a few modules chosen by the assessor.	Verify that procedures state that code must be reviewed and that hardware implementation must be reviewed. Verify that a security checklist exists and must be used as part of the review and that the checklist contains items (a), (b), (c), and (e) from the requirement at a minimum.. Pick a project that was developed using the same process being evaluated and verify that some code has been reviewed for that project, and that there is a clear list of which code has been reviewed. Verify there is some evidence that the code review checklist was used during the review (such as a completed checklist or a statement about the checklist in the code review results). In order to verify that the code has been reviewed, you may verify that the code review results are documented along with the following information: name of the person who performed the code review, the date of the code review, the results of the code review and the name of the person responsible for fixing problems identified in the code review and a date or indication that all problems were fixed. Code review results can be documented electronically or via paper copies, but the results must be available to an auditor. Items identified in the code review that were not fixed should be identified along with an explanation as to why they were not fixed. The code review results should be inspected for a few modules chosen by the assessor.	SDLA-MIV-2	
					SDLA-SI-1A	Verify that the list of code that has been reviewed includes all code which meets the stated criteria. This requirement does apply to legacy code but does not apply to third party embedded code.	Verify that documented procedures define a criteria for when an implementation review is required. Verify that the criteria is based on a risk analysis identifying which modules have the highest security risk.	SDLA-MIV-2.2	
X	X				SDLA-SI-1C-1	Determine if static analysis tools are available for the languages used. For those cases where such tools are available, verify that static analysis has been run on all source code (excluding third party embedded code) that meets the stated criteria and that the results have been documented. Certifier may elect to witness the supplier running static code analysis on certifier-selected portions of the code if they judge this activity required for sufficient confidence in any aspects of the validations on this topic. In this case, during test witnessing the certifier may inspect the tool configuration against process documentation and test report documentation (such as types of errors scanned for vs. coding standards, and portions of code excused from the scan), the versions of code analyzed against process requirements, and the witnessed results against the test report. *Witnessing* in the present context means that the certifier requests a new execution and resulting artifacts from a test process presented as evidence for the certification evaluation. The certifier may determine whether they will be physically present for some or all of the execution of the process.	Verify that the development procedures state that security static analysis tools (if available for the language used) should be run on all source code that meets criteria that is defied in the development process and that the results must be documented. Verify that the documented development process defines the criteria used to determine which source code is subject to static analysis, and that at a minimum the following is included: -Code listening on or connecting to a network that may be connected outside the Security Zone of the device, system or application under consideration -Code with prior vulnerabilities identified -Code executing with high privilege (for example SYSTEM, administrator, root) unless all code executes with high privilege -Security related code (for example, authentication, authorization, cryptographic and firewall code) -Code that parses data structures from potentially untrusted sources -Setup code that set access controls or handles encryption keys or passwords -All new code written after this procedure was put into place. Pick a project that follows the same development procedure being evaluated and verify that security static analysis tools have been run on some source code and that the results have been documented. Note: Third party included source code may be excluded from the static analysis requirements.	SDLA-MIV-3.1	This validation activity also covers requirement SI-2c which talks about automated tools used to determine if secure coding standards are being followed.
					SDLA-SI-1C-2	For those cases where such tools are available, review several changes made during the release being evaluated and verify that security static analysis tools have been run on this code (excluding third party embedded code) and that the results have been documented.			
					SDLA-SI-1C-3	None Required	Verify that evidence exists showing that most of the potentially exploitable coding constructs identified in the coding guidelines are checked for by the static analysis tool. User documentation of the tool along with a customer description on how the tools is setup and used is considered sufficient evidence if the tool is a well known commercially available tool. If the tool is developed in house, testing is required as evidence that the tool detects most potentially exploitable coding constructs from the security coding standard.	SDLA-MIV-3.2	
					SDLA-SI-2	Verify that coding standard is being followed by reviewing artifacts such as code review minutes or static analysis results or by looking at code.	Verify that a security coding standard is documented and that there is a process in place to ensure that it is followed. This process can consist of using static analysis to enforce the security coding standard, manual code review or some combination of both. Pick a project that has been developed with the same process being evaluated and verify that the coding standard is being followed by reviewing artifacts such as code review minutes or static analysis results or by looking at code.	SDLA-MIV-1	The security coding standard does not have to be an independent document. It may, for example, be part of an overall coding standard.

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SI-2	Secure coding standards	The implementation processes shall incorporate security coding standards that are periodically reviewed and updated and include at a minimum: a) avoidance of potentially exploitable implementation constructs – implementation design patterns that are known to have security weaknesses; b) avoidance of banned functions and coding constructs/design patterns – software functions and design patterns that should not be used because they have known security weaknesses; c) automated tool use and settings (for example, for static analysis tools); d) secure coding practices; e) validation of all inputs that cross trust boundary. f) error handling	SDLA-SI-2A	None Required	Verify that the documented security coding standard includes a list of potentially exploitable coding constructs or designs that should be avoided. Determine the basis of this part of coding standard and verify that it is from a recognized source based on real world security attacks. The following sources should be considered: The CERT secure coding standards, OWASP Secure Coding Practices, Common Weakness Enumeration (CWE), Microsoft Secure Coding Guidelines, or the SANS Top 25 Most Dangerous Software Errors. If one of these sources is not used, the coding standard should be comparable to these secure coding standards. This can be shown, for example, by documenting how the coding standard addresses the CWE or the SANS Top 25 list.	SDLA-MIV-1.2	
					SDLA-SI-2B	None Required	Verify that the documented security coding standard includes banned functions.	SDLA-MIV-1.3	Common C library functions such as strcpy(), gets(), and strcat() are highly susceptible to security problems which can be corrected by using alternate functions with built in checking such as strncpy(), fgets(), and strncat().
					SDLA-SI-2D	None Required	Verify that the documented security coding standard includes secure coding practices that should be followed. This can be done by reviewing the coding standard and verifying that there are specific items listed as secure coding practices. These practices should be based on techniques used to avoid problems that are known to lead to vulnerabilities. It should include techniques from well known sources such as CERT C coding standard.		Note: SI-4C is covered by the validation activity for SDLA-SI-1C-1
					SDLA-SI-2E	Inspect the detailed component or system design specification and verify that it documents where input validation testing will be done and the details of that validation. Verify that reviews of the design were held and the reviews checked for adequate input validation (i.e. completed checklist or this check explicitly mentioned in meeting minutes)	Verify that the software development process or design review checklist states that input validation must be done wherever data can enter the system or cross a trust boundary.	SDLA-DSD-3	
					SDLA-SI-2F	None Required	Verify that the documented coding standard includes guidelines for error handling.		
					SDLA-SI-2G	None Required	Verify that the documented process requires a periodic update of commonly accepted security recommended practices and coding guidelines based on commonly accepted practices in industry and lessons learned from vulnerabilities found in product. This can be verified if a documented procedure can be shown stating that these practices should be periodically updated. The procedure should state that the periodic update is based on some well known industry standards and guidelines. In addition, there should be a documented process to analyze security vulnerabilities that escape to the field (This is covered in requirement SM-13, no need to revisit here). Verify that this process, as reviewed in SM-13, is applied to the security recommended practices and coding guidelines as documented in the security coding standard.		
X		N/A	Applicability to systems level code.	The requirements of this phase that are applicable to system development, shall only apply to code written in a full variability language.	SDLA-SI-3	Verify whether a full variability language was used. If so, all requirements with the "System" column checked apply. If no requirements can be marked as not applicable.	Verify whether a full variability language was used. If so, all requirements with the "System" column checked apply. If no requirements can be marked as not applicable.	SDLA-MIV-6	A full variability language is one with full flexibility used to define a particular application . A limited variability language is a type of language that provides the capability to combine predefined, application specific, library functions to define a particular application. C, C++ and Java are examples of full variability languages. Function blocks and ladder logic are examples of limited variability languages.

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SVV-1	Security requirements testing	A process shall be employed for verifying the product security functions meet the security requirements and that the product handles error scenarios and invalid input correctly. Types of testing shall include: a) functional testing of security requirements; b) performance and scalability testing c) boundary/edge condition, stress and malformed or unexpected input tests not specifically targeted at security; and d) trust boundary requirements testing	SDLA-SVV-1A1	Verify that a security validation test plan is created or that the general validation test plan has a section for security. Verify through sampling that all security requirements have test cases associated with them.	Verify that the development process states that a security validation test plan must be created or that the general validation test plan must have a section for security. Verify that the development process states that the validation test plan must include tests to verify all security functions defined the security requirements work properly. Verify that this was done for a product developed using the process under evaluation. Verify that this testing is required for every release of software (although it is not required that all requirements are tested for every release of software). It is acceptable if a set of requirements that must be tested is created for each version of software based on what changed in that version (for example a security patch may have a smaller set of tests run against it than a major release would).	SDLA-SVT-1	
					SDLA-SVV-1A2	Verify that the validation results show that the plan was executed. This can be done by looking for references to the plan and verifying a subset of the results to make sure that what was done matches the plan.	Verify that the development process states that validation must be carried out as specified in the validation plan.	SDLA-SVT-2	
					SDLA-SVV-1A3	Verify that the validation results are documented.	Verify that the development process states that validation results must be documented. Verify that this was done for a product developed using the process under evaluation.	SDLA-SVT-3	
					SDLA-SVV-1B	Verify that performance and scalability testing was carried out for the product or system being evaluated.	Verify that the development process states that performance and scalability testing is required.		
					SDLA-SVV-1C	Verify that this type of testing has been done on the product or system being evaluated by looking for evidence such as a completed checklist or review meeting minutes showing that this was reviewed. Finding evidence in specific test plans may be done as well, but this is not sufficient by itself because you must verify that this is done as a normal part of the process rather than in just one instance.	Verify that the development process states that boundary/edge condition, stress and malformed or unexpected input tests are part of standard testing. Verify that there is some sort of checklist or review process the ensures that this occurs.		This item covers both (c) and (d). Malformed and unexpected input tests are done at the trust boundaries.
X	X	SVV-2	Threat Mitigation Testing	A process shall be employed for testing the effectiveness of the mitigation for the threats identified and validated in the threat model. Activities shall include: a) creating and executing plans to ensure that each mitigation implemented to address a specific threat has been adequately tested to ensure the mitigation works as designed and b) creating and executing plans for attempting to thwart each mitigation.	SDLA-SVV-2-1	Verify that there is evidence that all threats in the threat model that have been mitigated are included in the abuse case test plan. This can be shown by creating a traceability matrix that shows which threats are covered by which tests. Sample some of these tests and verify that they include attempts to thwart the mitigation as well unless this is not practical for a given mitigation. For cases where it is not practical, this should be explicitly stated so this can be differentiated from the case where it was forgotten.	Verify that the development process states that abuse case testing shall attempt to exploit all threats identified in the threat model that have been mitigated. Verify that the development process also states that attempts to thwart the mitigation must be included.	SDLA-SIT-2.1 SDLA-SIT-2	
					SDLA-SVV-2-2	Inspect test results and verify that they include all of the information documented in the requirement, and that all tests ultimately passed.	Verify that the development process states that abuse case test results must be documented. Pick a product that is developed using the process under evaluation and verify that abuse case test results were documented.	SDLA-SIT-2.2 SDLA-SIT-2.3	
X	X				SDLA-SVV-3A1	Verify that a fuzz test plan exists and verify that the fuzz test plan covers all interfaces that parse data sent to the component or system (where tools are available).	Verify that the development process states that a fuzz test plan must be created and must include fuzz testing of all interfaces that parse external data sent to the component or system (if a tool is available for that interface). Pick a project developed using the same process being evaluated and verify that a fuzz test plan exists, and includes all of the information documented in the requirement. Note: For custom protocols that run over TCP/IP, there are tools available that allow you to fuzz those protocols, but you have to feed information into the tool about the protocol description. For this type of scenario, where there is no tool that was developed specifically for a protocol, but there are tools that can be customized for the protocol, it shall be considered that a tool is available and therefore this requirement does apply).	SDLA-SIT-1.1 SDLA-SIT-1	Dumb fuzzing involves randomly corrupting data. Smart fuzzing involves analyzing the data and intelligently corrupting it with invalid, out of range, and other values. Grammar fuzzing is an example of smart fuzzing.
					SDLA-SVV-3A2	Review the fuzz test artifacts and verify that they demonstrate that the quality requirements in Appendix B (SDLA-SVV-3A2-A through F) have been met.	None Required		This requirement is needed in order to ensure that the fuzz testing is effective. In order to be effective, fuzz testing needs to include either some intelligence or many test cases. For example, if a message has a CRC on it, and the fuzzer is not calculating the CRC, then close to 100% of all messages will be rejected by the CRC and the test may only be an effective test of the CRC check.

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
		SVV-3	Vulnerability testing	A process shall be employed for performing tests that focus on identifying and characterizing potential security vulnerabilities in the product. Known vulnerability testing shall be based upon, at a minimum, recent contents of an established, industry-recognized, public source for known vulnerabilities. Testing shall include: a) abuse case or malformed or unexpected input testing focused on uncovering security issues. This shall include manual or automated abuse case testing and specialized types of abuse case testing on all external interfaces and protocols for which tools exist. Examples include fuzz testing and network traffic load testing and capacity testing. b) attack surface analysis to determine all avenues of ingress and egress to and from the system, common vulnerabilities including but not limited to week ACLs, exposed ports and services running with elevated privileges. c) black box known vulnerability scanning focused on detecting known vulnerabilities in the product hardware, host or software components. For example, this could be a network based known vulnerability scan. d) for compiled software, software composition analysis on all binary executable files, including embedded firmware, delivered by the supplier to be installed for a product. This analysis shall detect the following types of problems at a minimum: 1) known vulnerabilities in the product software components; 2) linking to vulnerable libraries; 3) security rule violations; and 4) compiler settings that can lead to vulnerabilities. e) dynamic runtime resource management testing that detects flaws not visible under static code analysis, including but not limited to denial of service conditions due to failing to release runtime handles, memory leaks and accesses made to shared memory without authentication. This testing shall be applied if such tools are available.	SDLA-SVV-3A3	Verify that a plan for network traffic load testing exists and covers all network interfaces that parse data sent to the component or system.	Verify that the development process states that a network traffic load (flooding) test plan must be created and must include testing of all network interfaces that parse external data sent to the component or system. Pick a project developed using the same process being evaluated and verify that a network traffic load test plan exists, and includes all of the information documented in the requirement.		
					SDLA-SVV-3A4	Review the network traffic load test artifacts and verify that they demonstrate that the quality requirements in Appendix B (SDLA-SVV-3A4-A through F) have been met.	None Required		
					SDLA-SVV-3A5	Inspect test results that meet SVV-3A and verify that they include all of the information documented in the requirement, and that all tests ultimately passed.	Verify that the development test states that fuzz test results must be documented. Pick a product that is developed using the process under evaluation and verify that fuzz test results were documented for that product.	SDLA-SIT-1.3	
	X				SDLA-SVV-3B	Verify that that attack surface analysis testing is performed for the component being evaluated . Verify that if a tool exists for the platform that the component runs on, then the tool is used to assist in this testing. Verify that the person doing the testing has training or experience in how to find these types of problems. Verify that these tests are documented as part of a test plan and test results.	Verify that the development process states that attack surface analysis testing must be performed and documented. Verify that the process states that if a tool exists for the platform that the component runs on, then the tool should be used to assist in this testing. Verify that the competency requirements for this tester are documented.		
X	X				SDLA-SVV-3C1	Verify that an known vulnerability detection test plan exists and includes all of the items described in the requirement.	Verify that the development process states that a known vulnerability detection test plan shall be created. Pick a product that is developed using the process under evaluation and verify that a known vulnerability detection test plan was created.	SDLA-SIT-3 SDLA-SIT-3.1	
X	X				SDLA-SVV-3C2	Inspect test results and verify testing was performed just prior to release, that the test results include all of the information documented in the test plan and that all tests ultimately passed.	Verify that the development process states that known vulnerability detection test results must be documented. Pick a product that is developed using the process under evaluation and verify that known vulnerability detection test results were documented.	SDLA-SIT-3.2	
	X				SDLA-SVV-3D	Look for evidence that binary composition analysis has been done on the component being evaluated if a tool for doing this exists on the platform of the component. The evidence should take the form of a test plan and test results documents which show that this testing was planned and carried out and test results were documented. Verify that any issues found were assessed and addressed as defined in their standard process (See DM-4). Verify by looking at the tool user documentation, that the tool can detect the following types of problems: 1) known vulnerabilities in the product software components, 2) linking to vulnerable libraries, 3) security rule violations, and 4) compiler settings that may lead to vulnerabilities	Verify that the development process states that binary composition analysis is required if a tool for doing this analysis exists on the platform of the product. Look for evidence that this has been done on one or two projects. The evidence should take the form of a test plan and test results documents which show that this testing was planned and carried out and test results were documented.		
	X				SDLA-SVV-3E	Look for evidence that dynamic runtime resource management testing has been done on the component being evaluated if a tool for doing this exists on the platform of the component. The evidence should take the form of a test plan and test results documents which show that this testing was planned and carried out and test results were documented. Verify that any issues found were assessed and addressed as defined in their standard process (See DM-4). Verify by looking at the tool user documentation, that the tool can detect the following types of problems: 1) denial of service conditions due to failing to release runtime handles, 2) memory leaks, 3) accesses made to shared memory without authentication	Verify that the development process states that dynamic runtime resource management testing is required if a tool for doing this analysis exists on the platform of the product. Look for evidence that this has been done on one or two projects. The evidence should take the form of a test plan and test results documents which show that this testing was planned and carried out and test results were documented.		

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
	X				SDLA-SVV-3	Certifier may elect to witness the supplier running on certifier-selected portions of the vulnerability tests under SVV-3 if they judge this observation required for sufficient confidence in any aspects of the validations for this requirement. In this case, during test witnessing the certifier may inspect the tool configuration against process documentation and test report documentation (such as types of vulnerabilities addressed and portions of code excused from the scan), the component configuration (such as adherence to recommended control system product install instructions and processing load), the versions of code tested against process requirements, and the witnessed results against test reports in evidence. *Witnessing* in the present context means that the certifier requests a new execution and resulting artifacts from a test process presented as evidence for the certification evaluation. The certifier may determine whether they will be physically present for some or all of the execution of the process. (Note that the certifier will always directly run a black box known vulnerability scan on the component or system as part of ISASecure certification, as required by CSA-300 and SSA-300 hence witnessing under SVV-3C of that testing is not expected.)	None Required		
X	X	SVV-4	Penetration Testing	A process shall be employed to identify and characterize security-related issues via tests that focus on discovering and exploiting security vulnerabilities in the product.	SDLA-SVV-4		Verify that the documented development process requires penetration testing to be performed. Verify that the results of this testing must be documented and that any issues found must be handled per the standard process for assessing and addressing security related issues (See SDLA-DM-4)		Penetration testing focuses specifically on compromising the confidentiality, integrity or availability of the product. It can involve defeating multiple aspects of the defence in depth design. For example, bypassing authentication to access the product, using elevation of privilege to gain administrative access and then compromising confidentiality by breaking encryption. As this example shows, penetration testing involves approaching testing like an attacker and often involves exploiting chained vulnerabilities in a product. This process is required to ensure that efforts have been taken to discover security-related issues in the product or product documentation that could allow the product to be exploited. Having this process means that the product supplier attempts to breach the security of the product through penetration testing. Penetration testing consists of confirming that vulnerabilities in any product capability or the defence in depth strategy can be exploited and used to compromise security of the product. It requires in depth knowledge of the product along with security testing tools and techniques. Penetration testing may involve the use of manual techniques, test tools or combinations of the two.
X	X	SVV-5	Independence of Testers	A process shall be employed to ensure that individuals performing testing are independent from the developers who designed and implemented the product according to the following table (see next row). The levels of independence are defined 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.	SDLA-SVV-5		Verify that the documented development process requires independence of testers consistent with table below.		

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications

Test type	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 person
Known vulnerability scanning	SVV-3 – Vulnerability testing	Independent person
Software composition analysis	SVV-3 – Vulnerability testing	None
Penetration testing	SVV-4 – Penetration testing	Independent department or organization

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	DM-1	Receiving notifications of security-related issues	A process shall exist for receiving and tracking to closure security-related issues in the product reported by internal and external sources including at a minimum: a) security verification and validation testers, b) suppliers of third-party components used in the product, c) product developers and testers, and d) product users including integrators, asset owners, end users and maintenance personnel NOTE External security verification and validation testers include researchers	SDLA-DM-1A	Not Applicable.	Verify that the mechanism is made publicly available, for example on the company's web site.	SDLA-SRP-1 SDLA-SPV-1.8	Examples include a dedicated e-mail address or phone number to report potential security vulnerabilities.
					SDLA-DM-1B	Not Applicable.	Verify that a documented process exists, and that it requires tracking issues to closure. Review list of issues reported through this process (if there are any) and pick a few to analyze further to ensure that they were tracked to closure. If no issues have been reported through this process, use the method described in SDLA-DM1A and verify that it gets logged into the system and track to closure.	SDLA-SRP-2	
X	X	DM-2	Reviewing security-related issues	A process shall exist for ensuring that reported security-related issues are investigated in a timely manner to determine their: a) applicability to the product, b) verifiability, and c) threats that trigger the issue. NOTE Timeliness is driven by market forces.	SDLA-DM-2	Not Applicable.	Verify that the documented process includes this step.	SDLA-SRP-2.1	
X	X	DM-3	Assessing security-related issues	A process shall be employed for analyzing valid security-related issues in the product to include: a) assessing their impact with respect to: 1) the actual security context in which they were discovered, 2) the product's security context (Practice 2), and 3) the product's defence in depth strategy (Practice 3), b) severity as defined by a vulnerability scoring system (for example, CVSS) c) identifying all other products/product versions containing the security-related issue (if any), d) identifying the root cause of the issue, and e) identifying related security issues. For root cause analysis, a methodical approach such as that described in IEC 62740 [25] may be employed	SDLA-DM-3	Not Applicable.	Verify that the documented process includes analyzing security related issues, that a bug tracking system is in place, and that existing security vulnerabilities are assigned a severity or criticality.	SDLA-SRP-2.2	
					SDLA-DM-3C	Not Applicable.	Verify that the documented process includes identifying other products/product versions that contain the same security related issue as well as identifying related issues that may need to be addressed as well.	SDLA-SRP-2.4	"A related vulnerability may result from repeating the same mistake that caused the reported vulnerability in similar code or from an underlying design flaw that leads to a pattern of vulnerabilities"[1] Related vulnerabilities should be fixed if they are similar enough to the original problem that the attacker would be likely to try them. For example if there are other similar interfaces that have the same vulnerability, they should be addressed.
					SDLA-DM-3D	Not Applicable.	Verify that process states that root cause analysis must be done. Verify that it has been done for existing vulnerabilities (ones that were found after this step became part of the process).	SDLA-SRP-2.6	
					SDLA-DM-3A	Not Applicable.	Verify that the documented process calls for a creation of an impact analysis when changes may affect security. Audit some recent modifications that affected security to see if an impact analysis was done and documented. Verify that the impact analysis documents the security lifecycle phases to be repeated.	SSDA-SRP-4	
X	X	DM-4	Addressing security-related issues	A process shall be employed for addressing security-related issues and determining whether to report them based on the results of the impact assessment (DM-3 – Assessing security-related issues). The supplier shall establish an acceptable level of residual risk that shall be applied when determining appropriate way to address each issue. Options include one or more of the following: a) fixing the issue through one or more of the following: 1) defence in depth strategy or design change; 2) addition of one or more security requirements and/or capabilities; 3) use of compensating mechanisms; and/or 4) disabling or removing features b) creating a remediation plan to fix the problem, c) deferring the problem for future resolution (reapply this requirement at some time in the future) and specifying the reason(s) and associated risk(s), d) not fixing the problem if the residual risk is below the established acceptable level of residual risk In all cases the following shall be done as well: e) informing other processes of the issue or related issue(s), including processes for other products/product revisions, and f) inform third parties if problems found in included third-party source code When security related issues are resolved recommendations to prevent similar errors from occurring in the future shall be evaluated. This process shall include a periodic review of open security-related issues to ensure that issues are being addressed appropriately. This periodic review shall at a minimum occur during each release or iteration cycle. NOTE When the resolution decision is to fix the security-related issue in the product implementation, the timing of the release of the fix can result in a patch (see Practice 8) or the fix may be deferred until the next release.	SDLA-DM-4	View the list of security issues found during development. Verify that a severity was established for all issues and that all issues with a severity above the established level of residual risk were either fixed or addressed in some other manner. Also, verify that all issues of the appropriate severity have been addressed based on the required security level of the product as defined in the development organization verification activity defined for this requirement (e.g. if SL-C = 1, all critical issues identified are either corrected or the reason for them not being relevant has been documented).	Verify that the documented process includes this step. Verify that it applies to security issues found internally and externally throughout any phase of the development lifecycle. Verify that there is an established acceptable level of residual risk defined. Verify that the development process states deferring or not fixing the problem is only an option if the risk is less than the established acceptable level of residual risk. The threshold for acceptable risk varies by SL capability (SL-C) of the product and is defined using the base CVSS score as follows: SL-C = 1. All "critical" issues identified are either corrected or the reason for them not being relevant has been documented. SL-C = 2. All "critical" and "high" issues identified are either corrected or the reason for them not being relevant has been documented. SL-C = 3. All "critical", "high", and "medium" issues identified are either corrected or the reason for them not being relevant has been documented. SL-C = 4. All issues identified are either corrected or the reason for them not being relevant has been documented Verify that there is a periodic review of open issues. Verify that a documented mechanism exists to inform third party suppliers if errors are uncovered in their product.	SDLA-SRP-2.3	Depending on the severity of the vulnerability, the plan could be to do nothing, to issue a service memo, to do an immediate patch release, to update in the next minor release, to update in the next major release, etc.
X	X	DM-5	Disclosing Security Related Issues	A process shall be employed for informing product users about reportable security-related issues (see 10.5, DM-4 – Addressing security-related issues) in supported products in a timely manner with content that includes but is not limited to the following information: a) issue description, vulnerability score as per CVSS or a similar system for ranking vulnerabilities, and affected product version(s); and b) description of the resolution.	SDLA-DM-5	Not Applicable.	Verify that there is a documented process for informing product users about security related issues. Verify that there is evidence that this process has been followed and that the appropriate content from the requirement has been included. If no such issues have been identified, verify that user notification was at least considered during the assessment of one or more security issues that were reported either internally or externally, unless no such issues have been reported.	SDLA-SRE-3	

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	DM-6	Periodic review of security defect management practice	A process shall be employed for conducting periodic reviews of the security-related issue management process. Periodic reviews of the process shall, at a minimum, examine security-related issues managed through the process since the last periodic review to determine if the management process was complete, efficient, and led to the resolution of each security-related issue. Periodic reviews of the security-related issue management process shall be conducted at least annually.	SDLA-DM-6	Not Applicable.	Verify that there is a periodic review of the defect management process defined in the documented development procedures. If this is an SDLA renewal, verify that this review has occurred at least twice since the initial certification. Verify that the results of the review were documented and that recommended changes were tracked to closure.		

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 62443-4-1 IEC Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SUM-1	Security update qualification	A process shall be employed for verifying (1) security updates created by the product developer address the intended security vulnerabilities (2) security updates do not introduce regressions, including but not limited to patches created by: a) the product developer; b) suppliers of components used in the product; and c) suppliers of components or platforms on which the product depends. The Process should include a verification that update is not contradicting other operational, safety or legal constraints	SDLA-SUM-1	Not Applicable.	Verify that the development organization has a documented process in place to be notified when updates are available from third parties and to validate that all updates work properly with the suppliers products.	SDLA-SRE-5	
X	X	SUM-2	Security update documentation	A process shall be employed to ensure that documentation about product security updates is made available to product users that includes but is not limited to: a) the product version number(s) to which the security patch applies; b) instructions on how to apply approved patches manually and via an automated process; c) description of any impacts that applying the patch to the product, including reboot; d) instructions on how to verify that an approved patch has been applied; and e) risks of not applying the patch and mediations that can be used for patches that are not approved or deployed by the asset owner.	SDLA-SUM-2	For a system, if any components of the system presented for certification cannot be updated without replacing the component, verify that this information and related instructions are provided in the user documentation and certification report.	Verify that there is a documented process related to security update documentation and that it includes items (a) through (e) from the requirement. If any updates have been released to users under the development process being assessed, choose one update at random and verify that the required documentation was produced.		Some additional information that should be considered in the documentation includes the following: -the document # and revision of the security update document -reference to original 'security alert if applicable (alert indicating problem, but update not yet available) -the CVE # assigned to the vulnerability that this documentation (and update) are targeted to mitigate.
X	X	SUM-3	Dependent component or operating system security update documentation	A process shall be employed to ensure that documentation about dependent component or operating system security updates is made available to product users that includes but is not limited to: a) stating whether the product is compatible with the dependent component or operating system security update b) for security updates that are unapproved by the product vendor, the mitigations that can be used to in lieu of not applying the update.	SDLA-SUM-3	Not Applicable.	Verify that there is a documented process related to security update documentation and that it includes items (a) and (b) from the requirement. Determine whether any security updates have been released by vendors of components or platforms upon which a product depends, for any product in scope for the development process being assessed. If so, choose one dependent component or operating system security update at random and verify that the required documentation was produced.		
X	X	SUM-4	Security update delivery	A process shall be employed to ensure that security updates for all supported products and product versions are made available to product users in a manner that facilitates verification that the security patch is authentic.	SDLA-SUM-4	Verify that a method was used to assure users where the code came from and to verify that it has not been tampered with. If a method other than digital signing was used, verify that the method meets the intent of this requirement.	Verify that the development process states that a method must be used to assure users where the code came from and to verify that the code has not been tampered with since its publication.		Note: This is the same requirement as SM-6, but here it is applied only to security updates.
X	X	SUM-5	Timely delivery of security patches	A process shall be employed to define a policy that specifies the timeframes for delivering and qualifying (See SUM-1 – Security update qualification) security updates to product users and to ensure that this policy is followed. At a minimum, this policy shall consider the following factors: a) The potential impact of the vulnerability; b) Public knowledge of the vulnerability; c) Whether published exploits exist for the vulnerability; d) The volume of deployed products that are affected; and e) The availability of an effective mitigation in lieu of the patch.	SDLA-SUM-5	Not Applicable.	Verify that the supplier has a documented process in place in order to determine the timeframe required for delivery of security updates. Verify that factors (a) through (e) are considered in this process if they are applicable. Determine (1) whether any security updates have been released to users under the development process being assessed (2) whether any security updates have been released by vendors of components used in a product in scope for this development process (3) whether any security updates have been released by vendors of components or platforms upon which such a product depends. If so, examine a few of these cases and verify whether this policy has been followed.		

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SG-1	Product defence in depth	A process shall exist to create product user documentation that describes the security defence in depth strategy for the product to support installation, operation and maintenance that includes: a) security capabilities implemented by the product and their role in the defence in depth strategy; b) threats addressed by the defence in depth strategy; and c) product user mitigation strategies for known security risks associated with the product, including risks associated with legacy code.	SDLA-SG-1C	Inspect security guidelines for the component or system being evaluated and verify that they include known security risks. Verify mitigations to these risks have been included in the security guidelines as well. If no known security risks are documented, verify that none were identified during threat modeling, attack surface reduction or security design reviews.	Inspect security guidelines for a product developed with the development process being evaluated and verify that they include known security risks. Verify any mitigations to these risks made have been included in the security guidelines as well. If no known security risks are documented, verify that none were identified during threat modeling, attack surface reduction or security design reviews. Or verify that the development process states that security guidelines must contain this information.	SDLA-DSG-1.1.5	
					SDLA-SG-1B	Inspect security guidelines for the component or system being evaluated and verify that they include threats addressed by the defence in depth strategy.	Verify that there is a documented process for ensuring that threats addressed by the defence in depth strategy are included in the security guidelines.		
					SDLA-SG-1A	Inspect security guidelines for the component or system being evaluated and verify that they include security capabilities of the product.	Inspect security guidelines for a product developed with the development process being evaluated and verify that they include security capabilities of product. Or verify that there is a checklist or procedure that requires that security capabilities of the product are included in the security guidelines.		
X	X	SG-2	Defence in depth measures expected in the environment	A process shall be employed to create product user documentation that describes the security defence in depth measures expected to be provided by the external environment in which the product is to be used (see Clause 6, Practice 2 – Specification of security requirements). NOTE These measures can also come from DM-4 – Addressing security-related issues	SDLA-SG-2	Inspect the security guidelines for the component or system being evaluated and verify that they describe environmental requirements that must be satisfied by the user. If not, determine if any such requirements are needed.	Inspect security guidelines for a product developed with the development process being evaluated and verify that they describe environmental requirements that must be satisfied. Or verify that the development process states that security guidelines must contain this information. Or a security guidelines template or checklist indicates this information should be included in the security guidelines.	SDLA-DSG-1.1.1	An example environmental requirement is that if authentication of human users is provided by integration into a system level identification and authentication system as permitted by 62443-4-2 CR 1.1, the authentication process for accounts used for essential functions cannot rely upon a connection to an untrusted network. In case of such a dependency, the component could not comply with 62443-4-2 CSSC 1 which requires that access controls not prevent the operation of essential functions.
X	X	SG-3	Security hardening guidelines	A process shall be employed to create product user documentation that includes guidelines for hardening the product when installing and maintaining the product. The guidelines shall include, but are not limited to, instructions, rationale and recommendations for the following: a) integration of the product, including third-party components, with its product security context (see Clause 6, Practice 2 – Specification of security requirements); b) integration of the product's application programming interfaces/protocols with user applications; c) applying and maintaining the product's defence in depth strategy (see Clause 7, Practice 3 – Secure by design); d) configuration and use of security options/capabilities in support of local security policies, and for each security option/capability: 1) its contribution to the product's defence in depth strategy (see Clause 7, Practice 3 – Secure by design); 2) descriptions of configurable and default values that includes how each affects security along with any potential impact each has on work practices; and 3) setting/changing/deleting its value; e) instructions and recommendations for the use of all security-related tools and utilities that support administration, monitoring, incident handling and evaluation of the security of the product; f) instructions and recommendations for periodic security maintenance activities; g) instructions for reporting security incidents for the product to the product supplier; and h) description of the security best practices for maintenance	SDLA-SG-3A	Inspect the security guidelines for the system or component being evaluated and verify that they describe hardening guidelines, instructions and recommendations. that should be adhered to when installing the product or system.	Inspect security guidelines for a product developed with the development process being evaluated and verify that they outline the hardening guidelines, instructions and recommendations that should be adhered to when installing the product. Or verify that the development process states that security guidelines must contain this information.	SDLA-DSG-1.1.2	Best practices include setting up a firewall, documenting any risks people should know about the installation process, procedures for integrating with other products in a secure manner, properly handling upgrade scenarios, and locking down the software more securely than the default configuration.
					SDLA-SG-3B	If the product contains an API or a set of classes or objects that developers can use, verify that instructions, rationale, and recommendations for integrating user applications securely with the API are provided.	May inspect security guidelines for a product developed with the development process being evaluated and verify that if the product contains an API or a set of classes or objects that developers can use then instructions, rationale, and recommendations for integrating user applications securely with the API are provided. Or may verify that the development process states that security guidelines must contain this information.	SDLA-DSG-1.1.6	
					SDLA-SG-3C	Verify the existence of secure operation and maintenance instructions for the product or system being evaluated. Verify that these instructions describe the user responsibility for operating and maintaining the defence in depth strategy defined for the product or system	Verify that the development process states that secure operation and maintenance instructions must be created for each product. Verify that these instructions include best practices for maintenance and administration of the product.	SDLA-DSG-2	Addresses SG-3F and 3H as well.
					SDLA-SG-3D	Inspect the security guidelines and verify that they describe all security configuration options including default and recommended settings.	Inspect security guidelines for a product developed with the development process being evaluated and verify that they list and explain all security configuration options present in the system, and make note of their default and recommended settings. Or verify that the development process states that security guidelines for administrators must contain this information.	SDLA-DSG-1.1.2.1	When components or systems include third party components such as operating systems then the security setting of those third party components would be applicable to this requirement. In this case, it would be acceptable to reference third party documentation for default and recommended settings for those products. Any exceptions to the third party recommendations may be noted in the component or system security guidelines.
					SDLA-SG-3E	Determine if such tools exist, and if so verify that their usage is described in the security guidelines. Verify that if the tools themselves are not secure, the guidelines indicate that these tools should be removed from the system prior to completing the integration.	Verify that the development process states that security guidelines must include instructions on how to use any security tools that exist for the product. Or inspect security guidelines for a product developed with the development process being evaluated and verify that they describe how to use any security tools provided with the product.	SDLA-DSG-4	

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
				and administration of the product.	SDLA-SG-3G	Inspect security guidelines for users and administrators and verify that they contain procedures for reporting security vulnerabilities back to the product manufacturer.	Verify that the development organization has a published method for reporting security vulnerabilities back to the product manufacturer.	SDLA-DSG-1.2	
					SDLA-SG-3H	Inspect security guidelines and verify that they describe how to administer the product in a secure manner (unless the product does not have administrative capability)	May inspect security guidelines for a product developed with the development process being evaluated and verify that they include guidance that describes how to administer the product in a secure manner. Or may verify that the development process states that security guidelines must contain this information.	SDLA-DSG-1.1.3	
X	X	SG-4	Secure disposal guidelines	A process shall be employed to create product user documentation that includes guidelines for removing the product from use. The guidelines shall include, but is not limited to instructions and recommendations for the following: a) removing the product from its intended environment (Practice 2), b) including recommendations for removing references and configuration data stored within the environment, c) secure removal of data stored in the product, d) secure disposal of the product to prevent potential disclosure of data contained in the product that could not be removed as described in c) above	SDLA-SG-4	Verify that the security guidelines for the product or system being evaluated contain security disposal guidelines. Verify that the disposal guidelines address the following issues: a) removing the product from its intended environment (note, depending on the product, this may not have any security implications) b) including recommendations for removing references and configuration data stored within the environment, (this may or may not apply) c) secure removal of data stored in the product, (this usually involves destroying or erasing hard disks) d) secure disposal of the product to prevent potential disclosure of data contained in the product that could not be removed as described in c) above	Verify that the documented development process requires that secure disposal guidelines are required to be included in the security guidelines documentation. Verify that the process, or a checklist, or template includes the items (a) through (d) from the requirements.		
X	X	SG-5	Secure operation guidelines	A process shall be employed to create product user documentation that describes: a) responsibilities and actions necessary for users, including administrators, to securely operate the product; and b) assumptions regarding the behavior of the user/administrator and their relationship to the secure operation of the product.	SDLA-SG-5	Verify that operation instructions contain assumptions regarding the behavior of the user/administrator. This means that they should describe the best practices or recommend behavior of users and administrators while operating the product.	Covered by SG-3	SDLA-DSG-1 SDLA-DSG-1.1 SDLA-DSG-2 SDLA-DSG-1.1.4	
X	X	SG-6	Account management guidelines	A process shall be employed to create product user documentation that defines user account requirements and recommendations associated with the use of the product that includes, but is not limited to: a) user account permissions (access control) and privileges (user rights) needed to use the product, including, but not limited to operating system accounts, control system accounts and data base accounts; and b) default accounts used by the product (for example, service accounts) and instructions for changing default account names and passwords.	SDLA-SG-6	Verify that the security guidelines for the product or system being evaluated include information about user account permissions and privileges required to use the product as well as default accounts used by the product and instructions for changing usernames and passwords on these accounts.	Verify that a documented development process, or template or checklist indicates that the security guidelines must include information about user account permissions and privileges required to use the product as well as default accounts used by the product and instructions for changing usernames and passwords on these accounts.		

System	Component	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Number	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Name	ANSI/ISA-62443-4-1 IEC 62443-4-1 Requirement Description	SDLA ID	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Related SDLA v1 Requirements	Comments/Clarifications
X	X	SG-7	Documentation review	A process shall be employed to identify, characterize, and track to closure errors and omissions in all user manuals including the security guidelines to include: a) coverage of the product's security capabilities, b) integration of the product with its intended environment (Practice 2), and c) assurance that all documented practices are secure	SDLA-SG-7C	Verify that all user manuals were reviewed by security experts by reviewing meeting minutes and verifying that someone qualified as a security expert (Based on experience, education, or personal certification) was involved in reviewing each of the user manuals.	Verify that the development process states that all user manuals, including documented security guidelines and operation and maintenance instructions, should be reviewed by security experts to ensure that they do not document any insecure practices	SDLA-DSG-3	
					SDLA-SG-7A	Verify that there is evidence that the security guidelines were reviewed (such as meeting minutes or a review signoff). Verify that the review confirmed that all security capabilities are described in the security guidelines. This can be verified by a completed checklist, a comment in the meeting minutes or something similar.	Verify that the documented development process requires that the security guidelines are reviewed. Verify that there is a process or review checklist that indicates that the review should confirm that all security capabilities are described in the security guidelines		
					SDLA-SG-7B	Verify that issues found during the user manual reviews are documented and tracked to closure. This can all be documented in the meeting minutes, through an issue tracking system, or thorough a similar method.	Verify that the documented process requires that issues found during the security manual review are documented and tracked to closure.		

System	Component	SDLA v1 Requirement ID	SDLA v1 Requirement Name	SDLA v1 Requirement Description	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Source of Requirement	Comments/Clarifications
X	X	SDLA-SMP-5	CM System	The development organization shall have a Configuration Management (CM) process.	Verify that development organization has been shown to meet this requirement (See Development Organization and SDL Validation Activity Column).	Verify that a process is in place and documented to manage and control the configuration of the component or system, and changes to that configuration. Details of that process are documented and will be assessed in the child requirements.	ISO/IEC 15408-3: ALC_CMC.2.3C	
	X	SDLA-SMP-5.2	Ascertain Changes	The CM process shall provide an automated means to ascertain the changes between the current component and its preceding version.	Witness the automated generation of the list of changes between a current component and its previous version using.	Verify that a documented procedure exists to ascertain the changes between a current component or system and its previous version using an automated means. Verify that the procedure will create a list of differences between the current version and the previous version. The differences should include a list of all source code modules that have changed. And then for each module you should be able to see which lines of code have changed, and you should be able to see a side by side comparison showing added code, removed code, and changed code.	ISO/IEC 15408-3: ALC_CMC.5.9C	
X	X	SDLA-SMP-5.4	Component or System Identification	The CM process shall provide a reference (unique identifier) for the component or system which shall be unique to each version of the product.	Verify that a reference exists for each version of the component or system.	Verify that the CM procedure or plan states that each component or system will have a unique identifier.	IEC 61508-3: 6.2.3.c & ISO/IEC 15408-3: ALC_CMC.1.1D & ALC_CMC.1.1C	
	X	SDLA-SMP-5.4.1	Component Label	The current component shall be labeled with its reference.	Verify that a physical label documents the reference for a component or that the label can be retrieved electronically by the user.	Verify that the CM procedure or plan states that each component be labeled with its reference.	ISO/IEC 15408-3: ALC_CMC.1.1C	
X	X	SDLA-SMP-5.5	Authorized Changes	The CM process shall provide a means by which only authorized changes are made to the component or system, implementation representation, and to all other configuration items.	Verify that the mechanism to only allow authorized changes to be made to the component, or system is being used on the component or system being evaluated.	Verify that CM process has a mechanism to only allow authorized changes to be made to the component or system.	ISO/IEC 15408-3: ALC_CMC.3.4C & IEC 61508-3: 6.2.3.d & 6.2.1.o	The product implementation representation refers to all hardware, software, and firmware that comprise the physical product. In the case of a software-only product, the implementation representation may consist solely of source and object code.
X	X	SDLA-SMP-5.6	Modification Audit	The CM process shall support the audit of all modifications to a component or system's, configuration items, including the originator, date, and time in the audit trail.	Pick a few modifications, and verify that the CM process documents the originator, the date and time of the changes and that a mechanism exists to determine exactly what changed.	If possible, pick a few modifications to a product that is using this process, and verify that the CM process documents the originator, the date and time of the changes and that a mechanism exists to determine exactly what changed. If the process is new and it is not possible to view examples, verify that there is a written description of the process that describes how this requirement will be met.	ISO/IEC 15408-3: ALC_CMC.5.9C & IEC 61508-3: 6.2.3.e	
X	X	SDLA-SMP-5.7	CM System Evidence	The CM shall document evidence that the CM system is operating in accordance with the CM plan.	Review the CM plan and ask to see evidence that it is being followed for the component or system being evaluated.	Review the CM plan and ask to see evidence that it is being followed for any product.	ISO/IEC 15408-3: ALC_CMC.3.8C	

System	Component	SDLA v1 Requirement ID	SDLA v1 Requirement Name	SDLA v1 Requirement Description	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Source of Requirement	Comments/Clarifications
X	X	SDLA-SMP-5.7.1	Configuration Items Effectively maintained	The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.	For a few randomly selected configuration items from the component or system under evaluation, ask to see evidence that these items are under configuration control in the CM system.	For a few randomly selected configuration items for any product, ask to see evidence that these items are under configuration control in the CM system.	ISO/IEC 15408-3: ALC_CMC.3.7C	
X	X	SDLA-SMP-6	Configuration Management Plan	The development organization shall create a Configuration Management (CM) plan that defines how configuration items will be managed.	Verify that a configuration management plan exists for the component or system under evaluation.	Verify that the CM process states that a CM plan that defined how configuration items will be managed must be created.	IEC 61508-3: 6.2.3.a & DO 178B: 4.3 & ISO/IEC 15408-3: ALC_CMC.3.5C	
X	X	SDLA-SMP-6.1	Automated CM Tools	The CM plan shall describe the automated tools used in the CM system.	Verify that the CM plan describes the automated tools used in the CM System.	Verify that the CM plan template includes a section to describe the automated tools used in the CM System. If there is no CM plan template, verify that the documented CM Process defines what should be included in the CM plan and this section is included.	ISO/IEC 15408-3: ALC_CMC.4.4C & ALC_CMC.4.5C	
X	X	SDLA-SMP-6.2	CM Tools Usage	The CM plan shall describe how the CM system is used including how any automated tools (if applicable) are used in the CM system.	Verify that the CM plan describes how each automated tool (if applicable) is used in the CM System and how the overall system is used.	Verify that the CM plan template includes a section to describe how each automated tool is used in the CM System (if applicable) and how the overall system is used. If there is no CM plan template, verify that the documented CM Process defines what should be included in the CM plan and this section is included.	ISO/IEC 15408-3: ALC_CMC.3.6C	
X	X	SDLA-SMP-6.3	Stage for formal configuration control	The CM plan shall document the stage in the lifecycle at which formal configuration control is implemented.	Verify that the stage at which formal configuration control is implemented is documented in the CM plan.	Verify that the stage at which formal configuration control is implemented is documented in the CM plan template or in the CM Process documentation.	IEC 61508-3: 6.2.1.o	
X	X	SDLA-SMP-6.4	Acceptance Plan	The CM plan shall include an acceptance plan which shall describe the procedures used to accept modified or newly created configuration items as part of the component or system.	Verify that an acceptance plan exists and was followed.	Verify that the CM process states there shall be an acceptance plan which shall describe the procedures used to accept modified or newly created configuration items as part of the component or system.	ISO/IEC 15408-3: 2005: ACM_CAP.4.13C & ACM_CAP.4.3C	The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized.
X	X	SDLA-SMP-7	Configuration List	The CM documentation shall include a configuration list of all configuration items that comprise the component or system, and will be controlled by the CM process.	Verify that a configuration list exists and that it includes all of the items that make up the component or system, including a unique identifier such as a part number and version number for each item.	Verify that the CM process states that a configuration list is created and that it includes all of the items that make up the component or system, including a unique identifier such as a part number and version number for each item.	IEC 61508-3: 6.2.1.o & ISO/IEC 15408-3: ALC_CMC.1.1D	
X	X	SDLA-SMP-7.1	Configuration Item Description	The configuration list shall describe the configuration items that comprise the component or system.	Verify that descriptions exist for each configuration item and that they are clear.	Verify that the CM process states that the configuration list must describe all of the configuration items that comprise the product or system.	ISO/IEC 15408-3: ALC_CMS.1.2C	
X	X	SDLA-SMP-7.2	Configuration Identification Method	The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the component or system.	May verify that the documented method or convention used to uniquely identify each configuration item has been followed.	Verify that the method or convention used to uniquely identify each configuration item is documented or that the CM process states that this method or convention must be documented throughout the lifecycle of the component or system.	ISO/IEC 15408-3: ALC_CMC.2.2C	

System	Component	SDLA v1 Requirement ID	SDLA v1 Requirement Name	SDLA v1 Requirement Description	**Component or System Validation Activity** (Applies for Component or System Certification)	**Development Organization and SDL Validation Activity** (Applies for SDLA Certification. Also applies if for Component/System if organization has not been previously SDLA Certified)	Source of Requirement	Comments/Clarifications
X	X	SDLA-SMP-7.3	CM System Identification	The CM process shall uniquely identify all configuration items that comprise the component or system.	Witness a demonstration as to how the CM system uniquely identifies configuration items for the component or system being evaluated. Verify that the demonstration shows that for a given release, you can find out all of the source code included in that release including which revision of each module has been included. Verify that that you can also find other configuration items, such as documentation associated with the release along with the document version numbers.	Witness a demonstration as to how the CM system uniquely identifies configuration items for any product. Verify that the demonstration shows that for a given release, you can find out all of the source code included in that release including which revision of each module has been included. Verify that that you can also find other configuration items, such as documentation associated with the release along with the document version numbers.	ISO/IEC 15408-3: ALC_CMC.2.3C	
X	X	SDLA-SMP-7.4	Configuration Item Inclusion	The list of configuration items shall include all of the following items (see sub-requirements).	Verify that sub-requirements have been met	Verify that sub-requirements have been met		
X	X	SDLA-SMP-7.4.1	Configuration Item Inclusion	The list of configuration items shall include all items that make up the implementation representation of the component or system.	Verify that sub-requirements have been met	Verify that sub-requirements have been met	ISO/IEC 15408-3: ALC_CMS.3.1C	The product implementation representation refers to all hardware, software, and firmware that comprise the physical product. In the case of a software-only product, the implementation representation may consist solely of source and object code.
X	X	SDLA-SMP-7.4.2	CM of Design Documentation	The list of configuration items shall include all security design documentation including requirements specifications, design specifications, test plans and the security management plan.	Pick a few key security design documents pertaining to the component or system being evaluated and verify that they are managed by the configuration management system.	Verify that the CM process states that all security design documentation must be managed by the configuration management system. May pick a few key security design documents pertaining to any component using this CM process and verify that they are managed by the configuration management system.	ISO/IEC 15408-3: ALC_CMS.3.1C	
X	X	SDLA-SMP-7.4.3	Security Flaws	The list of configuration items shall include identified security flaws.	Verify that security flaws of the component or system are controlled by the CM system which can consist of many tools such as a version control tool and a problem reporting and tracking tool	Verify that the CM process states that security flaws of the component or system are controlled by the CM system which can consist of many tools such as a version control tool and a problem reporting and tracking tool	ISO/IEC 15408-3: ALC_CMS.4.1C	Any security flaws found in the product (i.e. vulnerabilities) should be documented in the CM system, most likely in the change management/change request tool. Flaws can be stored in separate system or database that is not released to customers.
X	X	SDLA-SMP-7.4.4	Development Tools	The list of configuration items shall include all development tools.	Verify that development tools are controlled by the CM system.	May verify that the CM process states that development tools are controlled by the CM system	ISO/IEC 15408-3: ALC_CMS.5.1C	

System	Component	SDLA ID of Parent Requirement(s)	SDLA ID	SDLA Requirement Name	Component or System Validation Activity (Applies for Component or System Certification)	**Comments/Clarifications**
X	X	SDLA-SVV-3A2 SDLA-SVV-3A4	SDLA-SVV-3A2-A SDLA-SVV-3A4-A	Fuzz and network traffic load testing - adequate maintenance of essential functions	For fuzz testing and network traffic load testing, verify that the test plan defines monitoring criteria which determine whether or not essential functions are adequately maintained by the product during testing. Verify that if the product does not meet these criteria during the test, the test plan defines the test as failed. Verify that the test report shows this monitoring and pass/fail criterion were carried out. The supplier definition for adequate maintenance of essential functions should consider the guidance in section 11.3.2 of IEC 62443-3-3 (for systems) and section 11.3.2 of IEC 62443-4-2 11.3.2 (for components) which state that safe operations should be maintained. An acceptable but not mandatory definition for "adequately maintained," is that (where present) control functions and the safety instrumented function are not affected under any network traffic conditions, and that any other essential functions are fully maintained except when the network interface used by a function is under network flood conditions. Further, these essential functions should recover without human intervention once flooding ceases.	For systems and components, maintenance of essential functions in a degraded mode during DoS events, is required by IEC 62443-3-3 SR 7.1 and IEC 62443-4-2 CR 7.1, respectively. Monitoring for essential functions during testing is required whether or not it is an integrated feature of the test tools used. See the informative Annex A to SSA-300 for one method of defining adequate maintenance of the control function.
X	x	SDLA-SVV-3A2 SDLA-SVV-3A4	SDLA-SVV-3A2-B SDLA-SVV-3A4-B	Fuzz and network traffic load testing - test under functional load	Verify in the test plan and test report that fuzz testing and network traffic load testing are carried out while the product is performing its IACS functions other than network communication, at the maximum load recommended to customers.	This test approach verifies that under DoS attack as simulated by these tests, the product maintains a possibly degraded mode for essential functions, under operating load conditions supported by the product.
X	X	SDLA-SVV-3A2 SDLA-SVV-3A4	SDLA-SVV-3A2-C SDLA-SVV-3A4-C	Fuzz and network traffic load testing - redundant configurations	Verify in the test plan and report, that where a test target may be configured redundantly (where there are two or more instances of the test target), fuzz and network traffic load testing includes the scenarios when all instances of the test target are operational and when one or more of the redundant test targets are not operational.	This test approach verifies that under DoS attack as simulated by these tests, the product maintains a possibly degraded mode for essential functions, under failover operating conditions supported by the product.
X	X	SDLA-SVV-3A2 SDLA-SVV-3A4	SDLA-SVV-3A2-D SDLA-SVV-3A4-D	Fuzz and network traffic load testing - Pre and post authentication test	Verify in the test plan and report, that for protocols with authentication, fuzz and network traffic load testing occurs both in a state before and after successful protocol-layer authentication.	

System	Component	SDLA ID of Parent Requirement(s)	SDLA ID	SDLA Requirement Name	Component or System Validation Activity (Applies for Component or System Certification)	**Comments/Clarifications**
X	X	SDLA-SVV-3A2	SDLA-SVV-3A2-E	Fuzz testing - test traffic characteristics	Verify for a sample subset of protocols supported by the component, that fuzz test cases include relevant cases under 1) - 8) enumerated below. To select the subset of protocols, the certifier selects for each fuzzing tool used by the supplier, at least one protocol that the supplier tested with that tool. At a minimum the subset of protocols must also include at least one proprietary protocol (if there are any) and at least one protocol to which each of 1) - 8) is relevant, if there are any. The certifier then verifies for each protocol in the sample subset, using tool documentation provided to the certifier by the IACS supplier, test tool configuration reports, and/or other evidence, that the tool or combination of tool(s) used for testing, includes test cases that meet or exceed the set of scenarios 1) - 8). 1) For each field of the tested protocol, values that violate message field constraints for permitted values or data type. 2) Where length, message type, or other message characteristics are provided dynamically in a protocol field, inconsistency of the message with the data in that field 3) Where a specified string is used to self-delimit a field, misuse or lack of use of that delimiter 4) Where field or message size are constrained by the protocol, violations of these constraints 5) Where fields present in a message are flexible, too few, too many, or incorrect ordering of fields 6) Incorrectly ordered, duplicated, or out-of context messages 7) Single anomalies of types 1-6 in messages are tested separately, at a minimum, though combinations may be tested 8) Message types valid for a protocol but known not supported by the product under test	A supplier may use several tools for broad fuzzing coverage on a protocol. Regarding 8, tests should show essential functions adequately maintained upon receipt of unsupported message types, as required by SDLA-SVV-3A2-A and SDLA-SVV-3A4-A. No other criteria for passing these supplier tests are specified here for the tests to be considered adequate for certification. Note that some tools may report a test as failed if an unexpected response or no response is returned from the device for an unsupported message type. This specification does not require that such a result be considered a test failure. Also as noted under SDLA-SVV-3A2-A and SDLA-SVV-3A4-A, monitoring of essential functions may or may not be integrated into pass/fail criteria employed by some test tools.
X	X	SDLA-SVV-3A2	SDLA-SVV-3A2-F	Fuzz testing - receipt of test traffic	For fuzz tests, verify based upon the test plan and/or test report that the test method includes assurance that the test traffic is received by the component under test.	Intervening switches and routers may remove malformed traffic used in fuzz testing.

System	Component	SDLA ID of Parent Requirement(s)	SDLA ID	SDLA Requirement Name	Component or System Validation Activity (Applies for Component or System Certification)	**Comments/Clarifications**
X	X	SDLA-SVV-3A4	SDLA-SVV-3A4-E	Network traffic load testing - valid traffic rates	For network traffic load tests, verify in the test plan and report that this testing includes sending valid traffic representing each supported protocol to the component just below the designed rate limit (if the component is rate limiting) and at the full negotiated link rate. The traffic representing a supported protocol either uses that protocol, or uses the protocol for the highest layer of the protocol stack used by that protocol, for which traffic load testing tools are available.	A *rate limit* is a traffic rate at which a device invokes a mitigation against flooding attacks (IEC 62443-4-2 CR 7.1 RE(1)). The device is designed to handle traffic below the rate limit. The validation verifies such a design, and also verifies that testing covers the case in which an attacker utilizes full network bandwidth to execute a network traffic load DoS attack. Traffic should be valid because invalid traffic may be dropped and not create the intended load on the test target.
X	X	SDLA-SVV-3A4	SDLA-SVV-3A4-F	Network traffic load testing - connection flood	Verify in the test plan and report that network traffic load testing for connection-based protocols, includes attempting to overwhelm storage resources by initiating many connections.	For TCP, an example is a SYN flood.