Number	Phase Name	Description				
PH1	Security Management Process (SMP)	Process for planning and managing security development activities to ensure that security is designed into a component or system				
PH2	Security Requirements Specification (SRS)	Document customer driven security requirements, security features and the potential threats that drive the need for these features.				
PH3	Security Architecture Design (SAD)	Software or system architecture design for components or systems				
PH4	Security Risk Assessment and Threat Modeling (SRA)	Determine which components can affect security; Plan which components will require threat analysis, security code reviews and security testing.				
PH5	Detailed Software Design (DSD)	Component or system design down to the module or zone level following security design best practices				
PH6	Document Security Guidelines (DSG)	Create guidelines that users and administrators of the component or system must follow to ensure security requirements are met				
PH7	Module Implementation & Verification (MIV)	Implement design by writing code following secure coding guidelines. Ensure that software modules or zones are implemented correctly. Includes security code reviews, static analysis and module testing				
PH8	Security Integration Testing (SIT)	Perform security specific tests such as fuzz testing, abuse case testing and vulnerability identification testing				
PH9	Security Process Verification (SPV)	Independent assessment that all required component or system development processes have been followed				
PH10	Security Response Planning (SRP)	Putting a process in place to be able to quickly respond to security issues found in the field if and when they happen.				
PH11	Security Validation Testing (SVT)	Confirming that all security requirements have been met preferably by test or by analysis.				
PH12	Security Response Execution (SRE)	Responding to security problems in the field. Taking action to both preventative and corrective action.				

Revision History

Revision	Date	Changes
3.0	14.02.10	Initial version published to http://www.ISASecure.org

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level[1]	Comments/Clarifications
Project Management										
X	X	SDLA-SMP-1	Security Management Plan	A security management plan, which documents the plan for ensuring that security is addressed throughout the development lifecycle, shall be created as a stand alone document or as part of another plan, unless security management is already included as part of the standard software development lifecycle.	Verify a security management plan exists for the component or system	Verify that a security management plan is included as part of the standard development lifecycle and verify by reviewing examples from past or ongoing projects.	No	IEC-61508-3: 6.2.1	1, 2, 3, 4	
X	X	SDLA-SMP-1.1	Identification of responsibilities	The persons, departments and organizations which are responsible for carrying out and reviewing the applicable security related activities shall be documented.	Verify that all security related activities and that those responsible for carrying out the activities are listed in the project documentation.	Verify the standard development lifecycle requires that all security related activities and those responsible for carrying out the activities are documented in a security management plan.	No	IEC-61508-1: 6.2.1b	1, 2, 3, 4	
X	X	SDLA-SMP-1.2	Review of security management plan	If a security management plan is created it shall be reviewed by all those who are assigned responsibility in the plan.	Verify the existence of review minutes with a list of action items, all of which have been closed.	Verify that the security development lifecycle procedure requires a review of the security management plan.	No	IEC-61508-1: 6.2.3, 6.2.4, & DO-178B: 4.2.g	1, 2, 3, 4	
X	X	SDLA-SMP-1.3	Lifecycle Model	The development organization shall establish a life-cycle model to be used in the development and maintenance of the component or system. This model shall be documented.	Verify that the component or system was developed using the lifecycle model that is documented. This can be shown by the existence of all of the deliverables defined in the lifecycle.	Verify that the lifecycle model is documented and includes all of the required phases of the security development lifecycle.	No	IEC 61508-1: 6.2.1.c, DO-178B: 4.3 & ISO/IEC 15408-3: ALC_LCD.1.1D	1, 2, 3, 4	
X	X	SDLA-SMP-1.3.1	Lifecycle Model Details	The lifecycle model shall document the transition between software lifecycle processes by specifying: (1) The inputs to the process, including feedback from other processes, (2) Any integral process activities that may be required to act on these inputs, (3) Availability of tools , methods, plans, and procedures.	Verify that development organization has been shown to meet this requirement (See Development Organization and SDL Validation Activity Column).	Verify that lifecycle documentation documents the inputs to each phase, the process activities within the phase, and any tools, methods, plans or procedures that should be used in the phase	No	DO-178B: 4.3b	1, 2, 3, 4	
X	X	SDLA-SMP-1.3.2	Agile Lifecycle Model	The lifecyle model used may be an agile lifecyle in which multiple sprints (iterations) are done for each release.	None. This requirement does not have to be validated, but ensures that an agile lifecycle is acceptable.	None. This requirement does not have to be validated, but ensures that an agile lifecycle is acceptable.	No		1, 2, 3, 4	
X	X	SDLA-SMP-1.3.2.1	Sprint Requirements	If an agile lifecycle is used, security phases may be skipped during some sprints, but each security phase must be done in at least one sprint. In this case, the security phases to be practiced for each sprint shall be documented.	If an agile method is used, confirm that the security phases to be practiced for each sprint are documented, and each phase is practiced in at least one sprint.	If an agile method is used, confirm that the required security phases to be practiced for each sprint must be documented, and that is required that each phase is practiced in at least one sprint.	No		1, 2, 3, 4	
X	X	SDLA-SMP-1.4	Basic Security Training	All people involved in software development of a component or system, that has security concerns shall be given basic training in good security engineering practice and the applicable secure development process. In addition, software developers shall receive detailed training on common basic causes and mitigation techniques. System integration personnel shall receive training in network security administration/configuration techniques involved in a system. Testers shall receive training in security test techniques. The security management plan should document the security training plan for all those working on the software development. Evidence shall exist to show that those who have been trained have obtained the required knowledge from the training.	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 the development process states that for each product a list of required security training must be created and tracked. Verify that the required security training has been identified and that at least some developers have been trained.	No	CLASP: Institute security awareness program Microsoft: Stage 0: Education and awareness IEC 61508-1: 6.2.1.h	1, 2, 3, 4	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.
X	X	SDLA-SMP-1.5	Competence	Those involved in software development of a component or system with security requirements must be competent in carrying out the tasks assigned to them.	Verify that there is evidence of the competence of all people involved in software development of the component or system. This evidence can take the form of experience and qualifications and/or performance reviews.	Verify that company has review procedure to ensure the people are competently performing their jobs and receive all required training. Alternatively, verify that company has a procedure to ensure that all of those involved in software development of a component or system that has security concerns have a minimum required competency.	No	IEC 61508-1: 6.2.1.h	1, 2, 3, 4	
X	X	SDLA-SMP-1.6	Development Tools	The development organization shall identify all development tools (including versions) used to create the component or system, and document this information.	Verify that the development tools and version numbers are documented. This information may be included in the security management plan or it could be documented elsewhere.	Verify that there is a development procedure or template that requires that all of the development tools and version numbers are documented. This information may be included in the security management plan or it could be documented elsewhere.	No	ISO/IEC 15408-3: ALC_TAT1.1D & IEC 61508-3: 7.4.4.2	1, 2, 3, 4	NOTE: Unless otherwise notified the reference ISO/IEC 15408-3 hereafter designates the 2008 version, i.e. ISO/IEC 15408-3:2008.
X	X	SDLA-SMP-1.6.1	Development Tools Options	The development organization shall document the selected implementation-dependent options of the development tools.	Verify for each development tool listed or sampling of tools listed whether there are any implementation dependent options, and if so whether they have been documented.	Verify that development procedure requires that implementation dependent options of development tools are documented in the security management plan.	No	ISO/IEC 15408-3: ALC_TAT1.2D	3, 4	
X	X	SDLA-SMP-1.7	Revision of security management plan	The software, or system, planning process should provide a means to revise the security management plan throughout the lifecycle of the component or system.	Verify that the documented procedure to revise the security management plan has been followed if the plan has been updated.	Verify that a procedure is in place to revise the security management plan throughout the lifecycle of the component or system.	No	DO-178B: 4.2e	1, 2, 3, 4	
X	X	SDLA-SMP-2	Action Item Resolution	A process shall exist for ensuring that action items from security-related review meetings are documented and tracked to closure.	Verify that a documented procedure exists to document and track action items to closure. Randomly review meeting minutes (e.g. security requirements review meetings, code review meetings, etc.) related to the component or system being evaluated and identify action items and verify whether they were tracked to closure.	Verify that a documented procedure exists to document and track security-related action items to closure.	No	IEC-61508-1: 6.2.1.g	1, 2, 3, 4	
X	X	SDLA-SMP-3	Documentation of software releases	Software configuration management should formally document the release of all software for the component or system.	Verify that the latest release is documented via release notes, a software release memo or some other mechanism.	Verify that the development procedure states that a release is documented via release notes, a software release memo or some other mechanism.	No	IEC 61508-3: 6.2.3.f	1, 2, 3, 4	
Development Environment Security										

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level[1]	Comments/Clarifications
X	X	SDLA-SMP-4	Development Environment Security Documentation	The development organization shall produce development security documentation which shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality, availability and integrity of the component or system, design and implementation in its development environment.	Verify that development organization has been shown to meet this requirement (See Development Organization and SDL Validation Activity Column).	Verify that development security documentation exists, and covers physical procedure and personnel security measures at a minimum.	No	ISO/IEC 15408-3: ALC_DVS.1.1.D & ALC_DVS.1.1.C	2, 3, 4	
X	X	SDLA-SMP-4.1	Development Environment Security Evidence	The development security documentation described in SMP-4 shall provide evidence that these security measures were followed during the development and maintenance of the component or system.	Verify that measures defined in development security documentation are being followed-by reviewing evidence provided by developer.	Not Applicable	No	ISO/IEC 15408-3: 2005: ALC_DVS.1.2C	2, 3, 4	
Software Configuration Management										
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.	No	ISO/IEC 15408-3: ALC_CMC.2.3C	1, 2, 3, 4	
X	X	SDLA-SMP-5.1	Software Generation	The CM process shall provide an automated means to support the generation of the software.	Verify that the software can be generated in an automated fashion by witnessing this generation.	Verify that the software can be generated in an automated fashion by confirming that the procedure for doing so is documented.	No	ISO/IEC 15408-3: ACM_CMC.4.5C	3, 4	
	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.	Yes	ISO/IEC 15408-3: ALC_CMC.5.9C	3, 4	
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.	No	IEC 61508-3: 6.2.3.c & ISO/IEC 15408-3: ALC_CMC.1.1D & ALC_CMC.1.1C	1, 2, 3, 4	
	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.	No	ISO/IEC 15408-3: ALC_CMC.1.1C	1, 2, 3, 4	
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.	Yes	ISO/IEC 15408-3: ALC_CMC.3.4C & IEC 61508-3: 6.2.3.d & 6.2.1.o	2, 3, 4	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.	No	ISO/IEC 15408-3: ALC_CMC.5.9C & IEC 61508-3: 6.2.3.e	1, 2, 3, 4	
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.	No	ISO/IEC 15408-3: ALC_CMC.3.8C	2, 3, 4	
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.	No	ISO/IEC 15408-3: ALC_CMC.3.7C	2, 3, 4	
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.	No	IEC 61508-3: 6.2.3.a & DO 178B: 4.3 & ISO/IEC 15408-3: ALC_CMC.3.5C	1, 2, 3, 4	
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.	No	ISO/IEC 15408-3: ALC_CMC.4.4C & ALC_CMC.4.5C	3, 4	
X	X	SDLA-SMP-6.2	CM Tools Usage	The CM plan shall describe how the CM system is used including how the automated tools are used in the CM system.	Verify that the CM plan describes how each automated tool 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 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.	No	ISO/IEC 15408-3: ALC_CMC.3.6C	2, 3, 4	
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.	No	IEC 61508-3: 6.2.1.o	1, 2, 3, 4	
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.	No	ISO/IEC 15408-3: 2005: ACM_CAP.4.13C & ACM_CAP.4.3C	3, 4	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.	No	IEC 61508-3: 6.2.1.o & ISO/IEC 15408-3: ALC_CMC.1.1D	1, 2, 3, 4	
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.	No	ISO/IEC 15408-3: ALC_CMS.1.2C	1, 2, 3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level[1]	Comments/Clarifications
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.	No	ISO/IEC 15408-3: ALC_CMC.2.2C	1, 2, 3, 4	
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.	Witness a demonstration as to how the CM system uniquely identifies configuration items for any product.	No	ISO/IEC 15408-3: ALC_CMC.2.3C	1, 2, 3, 4	
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	No		1, 2, 3, 4	
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 all hardware, software and firmware that comprise the physical component or system, are included as configuration items for the component or system being evaluated.	May verify that CM process states that the list of configuration items shall include all items that make up the implementation representation of the component or system. Or, may verify that all hardware, software and firmware that comprise the physical component or system, are included as configuration items for any component or system using this CM process..	No	ISO/IEC 15408-3: ALC_CMS.3.1C	1, 2, 3, 4	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.	No	ISO/IEC 15408-3: ALC_CMS.3.1C	1, 2, 3, 4	
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	No	ISO/IEC 15408-3: ALC_CMS.4.1C	3, 4	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	No	ISO/IEC 15408-3: ALC_CMS.5.1C	3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
General Requirements										
X	X	SDLA-SRS-1	Security Requirements Specification	A security requirements specification (SecRS) shall be created to document all required security functions of the component or system.	Verify security requirements specification exists for component or system under evaluation and includes required security functions. 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. May verify that security requirements exist for any product developed under the process being certified.	No	IEC 61508-3: 7.2.2.11, CLASP: Document security-relevant requirements ISO/IEC 15408-3: ASE_REQ.1.1D & ASE_OBJ.1.1D	1, 2, 3, 4	The SecRS doesn't need to be single document. Many organizations create a security requirements section in other requirements and customer documents.
X	X	SDLA-SRS-2	Component or System Description	The developer shall provide an component or system, description as part of the SecRS	Verify SecRS includes a component or system, description	May verify that the SecRS template includes a section for a device or system description. Or, if no template exists, may verify that development process states that the security requirements must include a description of the component or system. May verify that a SecRS created for any component or system using the development process being certified includes a description of the component or system.	No	ISO/IEC 15408-3: ASE_INT.1.1C	1, 2, 3, 4	
X	X	SDLA-SRS-2.1	Scope of component or System	The component or system, description shall describe the component or system type and the scope and boundaries of the component or system, in general terms in both a physical and a logical way.	Verify the description includes a description of the component or system, and the scope and boundaries of the device in both a physical and logical way.	May verify the description includes a description of the component or system, and the scope and boundaries of the device in both a physical and logical way for any component or system developed under the process being certified.	No	ISO/IEC 15408-3: ASE_INT_1.7C	1, 2, 3, 4	
Operating Environment										
X	X	SDLA-SRS-3	Operating Environment	The SecRS shall include a statement of expected security environment as defined in following child requirements so that the impact on security can be assessed.	Verify SecRS includes a description of the operating 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.	No	CLASP: Specify operational environment, ISO/IEC 15408-3: ASE_SPD.1.4C	1, 2, 3, 4	
X	X	SDLA-SRS-3.1	Operating Environment Assumptions	The statement of security environment shall identify and explain any assumptions about the intended usage of the component or system, and the environment of usage that must be met by administrators in order for the product to be secure.	Verify SecRS identifies and explains assumptions about the intended usage of the product and the 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.	No	CLASP: Specify operational environment, ISO/IEC 15408-3: ASE_SPD.1.4C	1, 2, 3, 4	These are the features provided by site security policies that are independent of the product. Examples: limited physical access, employee screening
X	X	SDLA-SRS-3.2	Known or Presumed Threats	The statement of security environment shall identify and explain any known or presumed threats to the assets against which protection will be required either by the component itself, or system itself, or by its environment.	Verify known or presumed threats to the assets which protection will be required are documented. Verify that there is evidence that requirements were reviewed and known/presumed threats list was included in review (e.g. meeting minutes or inclusion in completed review checklist.)	May verify known or presumed threats to the assets which protection will be required are documented in SecRS for any component or system developed according to the process being evaluated. Or may verify that the development process or SecRS template states that known or resumed threats to the assets which protection will be required must be documented.	No	ISO/IEC 15408-3: ASE_SPD.1.1C	1, 2, 3, 4	Some examples of threats from common criteria shown below: DATA_FLOODING A malicious user may subject communications channel entering a domain to higher than expected levels of messages to the product resulting in potential denial of service or compromise of the operations performed within the domain. ADMIN_ERROR An administrator may incorrectly install or configure the product resulting in ineffective security mechanisms. AUDIT_COMPROMISE A malicious user may compromise audit records masking a user's action.
Security Requirements Content										
X	X	SDLA-SRS-4	Basic Security Functions	Required security functions/features that implement the required organizational security policies shall be included in the SecRS.	Verify that the SecRS includes security features and functions.	Verify that the development process or SecRS template states that security features and functions must be documented. Verify that the SecRS for any component or system developed according to the process being evaluated includes security features and functions.	No	CLASP: Document security-relevant requirements	1, 2, 3, 4	The source for these requirements could be the security business requirements or it could be based on standards such as ISA S99
X	X	SDLA-SRS-5	Security Assurance Level	Required security assurance level for the product should be included in the SecRS	Verify that the security assurance level is documented in the SecRS	May verify that the development process or SecRS template states that the security assurance level must be documented.	No	IEC 61508: 7.2.2.11 ISO/IEC 15408-3: 2005: ASE_REQ.1.3C	1, 2, 3, 4	Refer to ISA 99.01.01 for a definition of Security Assurance Level (SAL).
X	X	SDLA-SRS-6	Regulatory Requirements	Any security related regulatory requirements that the component or system must comply with should be included in the security requirements specification.	Verify that regulatory requirements are documented or that there are no applicable regulatory requirements	May verify that the development process or SecRS template states security related regulatory requirements must be documented.	No		1, 2, 3, 4	
Quality of Requirements										

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SRS-7	Security Requirements Detail	The security requirements specification (SecRS) shall be sufficiently detailed to allow the design and implementation to achieve the required integrity, and to allow a security evaluation to be carried out.	Verify evidence of requirements review and approval by software developers and security experts and those representing the customer perspective (e.g. meeting minutes) plus evidence that requirements were reviewed for these specific qualities (e.g. details in meeting minutes or completion of review checklist).	Verify that development process states that security requirements must be reviewed by software developers, security experts and those representing the customer perspective and that the results of this review must be documented. Verify that the development process or review checklist states that the requirements provide enough detail so that they can be implemented to achieve the required integrity.	No	IEC 61508-3: 7.2.2.3	1, 2, 3, 4	
X	X	SDLA-SRS-8	Security requirements clarity	The security requirements shall be expressed and structured such that they are clear, precise, unequivocal, verifiable by test, analysis or other means, maintainable, and feasible, but do not contain unnecessary design or verification detail.	Verify evidence that the requirements were reviewed for these specific qualities (e.g. details in meeting minutes or completion of review checklist).	Verify that the development process or review checklist states that the requirements are structured such that they are clear, precise, unequivocal, verifiable by test, analysis or other means, maintainable, and feasible, but do not contain unnecessary design or verification detail.	No	IEC 61508-3: 7.2.2.06.a, DO-178B: 5.1.2.e, f, & g, CLASP: Document security-relevant requirements	1, 2, 3, 4	
X	X	SDLA-SRS-9	Security Requirements Review	Developers shall review the requirements to ensure that they are adequately specified. During this review, the requirements should be analyzed for ambiguities, inconsistencies, and undefined conditions.	Verify evidence that the requirements were reviewed for these specific qualities (e.g. details in meeting minutes or completion of review checklist).	Verify that the development process or review checklist states that the requirements are analyzed for ambiguities, inconsistencies, and undefined conditions.	No	IEC 61508-3: 7.2.2.4 & 7.2.2.06.c, DO-178B: 5.1.2.a ISO/IEC 15408-3: ASE_REQ.1.6C	1, 2, 3, 4	
X	X	SDLA-SRS-10	Security Requirements Additional Review	Any changes to the requirements after the initial review are subject to an additional review using the same review criteria.	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 states that all changes to the requirements after the initial review are subject to an additional review using the same review criteria.	No	IEC 61508-3: 7.4.3.3	1, 2, 3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
	X	SDLA-SAD-1	Component Software Partitioning	The component software architecture design description shall be based on partitioning the component into components or subsystems.	Inspect the architecture design description and verify that the design partitions the component into a relatively small number of components or subsystems	Verify that the development process requires a software architecture design description that partitions the component into components or subsystems.	No	IEC 61508-3: 7.4.3.2.b DO-178B: 11.10.b & i ISO/IEC 15408-3: ADV_TDS.1.1D	1, 2, 3, 4	The developer is expected to describe the design of the product in terms of subsystems or components. The terms “subsystem” and "component" are used interchangeably to express the idea of decomposing the product into a relatively small number of parts. While the developer is not required to actually have “subsystems” or "components" , the developer is expected to represent a similar level of decomposition. For example, a design may be similarly decomposed using “layers”, “domains”, or “servers
	X	SDLA-SAD-2	Network Interfaces	The component software architecture design shall describe all external network interfaces. This description shall include the actors expected to interact with the device.	Inspect the component architecture design description and verify that the design shows all network interfaces to the device.	Verify that the development process or software architecture design template indicates that all external network interfaces and the actors expected to interact with the device must be documented as part of the software architecture design. Or, inspect the component architecture design description for any device developed with the process being evaluated and verify that the design shows all network interfaces to the device.	No	CLASP: Identify Resources and Trust Boundaries	1, 2, 3, 4	
	X	SDLA-SAD-2.1	Interface Descriptions	The component software architecture design shall describe available protocols, the purpose, and method of use of all interfaces to the component providing details of effects, exceptions and error messages, as appropriate.	Verify that all interfaces are documented including the purpose, method of use and that the level of detail is sufficient.	Verify that the development process or software architecture design template indicates that the software architecture design must describe available protocols, the purpose, and method of use of all interfaces to the component along with details of effects, exceptions and error messages, as appropriate. Or, inspect the component architecture design description for any device developed with the process being evaluated and verify that this information has been included..	No	ISO/IEC 15408-3: ACO_DEV.1.1C	1, 2, 3, 4	
	X	SDLA-SAD-3	Dataflows	The component software architecture design shall identify data flows, including direction of the data flow (e.g. read or write) and the initiator, between the component and entities that are external to the device and within the device.	Inspect the component software architecture design description and verify that the design identifies data flows between the component and entities that are external to the device and within the component.	Verify that the development process or software architecture design template indicates that data flows between the component and external entities as well as within the component must be documented as part of the software architecture design. Or, inspect the component architecture design description for any component developed with the process being evaluated and verify that the design identifies data flows between the component and external entities and within the component.	No	CLASP: Identify Resources and Trust Boundaries	1, 2, 3, 4	Sample data structures include: • Databases and database tables • Configuration files • Cryptographic key stores • ACLs • Registry keys • Web pages (static and dynamic) • Audit logs • Network sockets / network media • IPC, Services, and RPC resources • Any other files and directories • Any other memory resource
X	X	SDLA-SAD-4	Trust Boundaries	The component or system architecture design shall document trust boundaries	Inspect the component or system architecture design description and verify that trust boundaries are documented.	Verify that the 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.	No	CLASP: Identify Resources and Trust Boundaries Microsoft: Stage 4: Risk Analysis	1, 2, 3, 4	Trust boundaries are demarcation points that show where data moves from lower privilege to higher privilege
X	X	SDLA-SAD-5	Attack Surface	The component or system architecture design shall enumerate the attack surface which includes all possible entry points for an attacker.	Verify that the attack surface is identified and documented in the component or system architecture design description.	Verify that the development process or architecture design template indicates that the attack surface 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 the attack surface is documented.	No	CLASP: Identify Attack Surface, Microsoft: Stage 2: Define and Follow best design practices ISO/IEC 15408-3: ADV_FSP.1.3C, ADV_TDS.1.3.C	1, 2, 3, 4	Attack surfaces includes all external interfaces, protocols and executing code. Access control measures should include each entry point and protocol. Protocol documentation should include open network ports. For components, interfaces include places where the file system is touched, local UI elements, inter-procedural communication points and any public methods that can
X	X	SDLA-SAD-6	Attack Surface Reduction	Attack surface reduction techniques shall be practiced to minimize the number of available entry points	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.	No	Microsoft: Stage 2: Define and Follow Design Best Practices CLASP: Identify User Roles and Resource Capabilities	1, 2, 3, 4	Entry points shall be minimized to only those absolutely necessary. For components, the attack surface can be reduced by reducing the amount of code that executes by default, restricting the scope of who can access the code, restricting the scope of which identities can access the code, and reducing the privilege of the code. For systems, the attack surface can be reduced by reducing the number of entry points, applying access controls, filtering/inspecting protocols, minimizing configuration options, hardening system components, etc.
	X	SDLA-SAD-7	Semi-Formal Methods	The presentation of the component software architecture shall be semi-formal.	Verify that the component software architecture has been documented using some sort of restricted syntax language such as dataflow diagrams, state transition diagrams, etc. Verify that the description is clear and sufficiently explained.	Verify that the development process or software architecture design template indicates that the architecture design must be documented using a restricted syntax language such as data flow diagrams, state transition diagrams, etc. Or, inspect the component architecture design description for any device developed with the process being evaluated and verify that the design has been documented using a restricted syntax language.	No	ISO/IEC 15408-3: ADV_TDS.4.4C IEC 61508-3: 7.4.3	3, 4	Semi-formal is defined as expressed in a restricted syntax language with defined semantics. The language can be graphical or textual.

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SAD-8	Secure Design Best Practice	The component or system design process shall incorporate secure design best practices. This applies to all features, not just security features.	Verify that the development process has been followed in the area of secure best practices. Verify that some of the secure best practices defined in this requirement have been employed in the development of the component or system being evaluated.	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). Typical best practices include economy of mechanism, fail-safe defaults, complete mediation, open design, separation of privilege, least privilege, least common mechanism, and psychological acceptability. At least some of these practices should be included on the list of best practices.	No	Microsoft: Stage 2: Define and Follow best design practices	1, 2, 3, 4	
	X	SDLA-SAD-9	Security Tools	Security tools to help administrators set a secure configuration and audit against a secure baseline shall be considered as part of the security design.	Verify that such tools were considered during the design as documented by meeting minutes, a completed checklist, or the existence of such tools.	Review the standard software development process and verify that consideration of security tools is part of the design process.	No	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	1, 2, 3, 4	Security tools are recommended, but not required. It is required, however, that they are considered during the development of the product and an explicit decision on whether to include them or not is made.
X		SDLA-SAD-10	System Partitioning	The system architecture design description shall be based on partitioning the system into zones (if applicable), subsystems, devices and network connections.	Inspect the architecture design description and verify that the design partitions the system.	Verify that the development process states that the design shall be partitioned into zones (if applicable), subsystems, devices and network connections. Or inspect the system architecture design for any system developed with the process being evaluated and verify that the design partitions the system.	No	IEC 61508-3: 7.4.3.2.b DO-178B: 11.10.b & i ISO/IEC 15408-3: ADV_TDS.1.1D	1, 2, 3, 4	
X	X	SDLA-SAD-11	System Network Design	The system architecture design shall describe the architecture of the system from the perspective of communication between subsystems and devices. This description shall also describe the connections with devices external to the system (e.g., higher level systems, other systems, remote administrators, external devices, etc.), as well as interactions between the devices and subsystems that comprise the system.	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.	Verify that the development process or system architecture design template states that the design shall document how the system's devices and subsystems are connected, and how external actors are connected to the system. Or inspect the system architecture design for any system developed with the process being evaluated and verify that this information is documented.	No	CLASP: Identify Resources and Trust Boundaries	1, 2, 3, 4	
X		SDLA-SAD-12	External Communication Protocols	The system architecture design shall identify the protocols used to communicate between external actors and 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.	Verify that the development process or system architecture design template states that the system design shall identify the protocols used to communicate between external actors and the system. Or inspect the system architecture design for any system developed with the process being evaluated and verify that this information is documented.	Yes, conducted as part of System Robustness Testing (see ISASecure SSA-310)		1, 2, 3, 4	
X		SDLA-SAD-13	Internal Communication Protocols	The system architecture design shall identify the protocols used to communicate between the systems devices and subsystems.	Inspect the system architecture design and verify that the design shows all protocols, used by all devices and subsystems, and over which connections the protocols are used.	Verify that the development process or system architecture design template states that the system design shall identify the protocols used to communicate between the systems devices and subsystems. Or inspect the system architecture design for any system developed with the process being evaluated and verify that this information is documented.	Yes, conducted as part of System Robustness Testing (see ISASecure SSA-310)		1, 2, 3, 4	
X		SDLA-SAD-14	System Dataflows	The system architecture design shall identify data flows that may be used or passed by the components of the system or with external entities as well as the direction of data flow (e.g. read or write).	Inspect the architecture design description and verify that the design identifies all data flows used in the design	Verify that the development process or system architecture design template states that the system design shall identify the data flows that may be used or passed by the components of the system or with external entities as well as the direction of data flow. Or inspect the system architecture design for any system developed with the process being evaluated and verify that this information is documented.	No	CLASP: Identify Resources and Trust Boundaries	1, 2, 3, 4	Applicant must identify dataflows external to SUT, zone to zone, within zones. Sample data flows include: • Databases and database tables • Configuration files • Web pages (static and dynamic) • Audit logs • Any other files and directories

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SRA-1	Security Design Reviews	During this phase, it shall be determined what portions of the project will require security design reviews.	Verify that a plan for security design reviews is documented	Verify that the development process requires that security design reviews be performed on some parts of the project. Verify that security design reviews have been done for any product or system that has been developed according to the same process being evaluated.	No	Microsoft Stage 3: Product Risk Assessment	1, 2, 3, 4	
X	X	SDLA-SRA-2	Required Abuse Case Testing	During this phase, it shall be determined what portions of the project will require abuse case testing	Verify that an abuse case test plan is documented	Verify that abuse case testing is required as part of the development process. Verify that an abuse case test plan was created for any product or system that has been developed according to the same process being evaluated.	No	Microsoft Stage 3: Product Risk Assessment ISO/IEC 15408-3: AVA_VAN.1.3E	1, 2, 3, 4	Abuse cases describe the system's behavior under attack. Abuse case tests are simulated attacks often based on the threat model.
X	X	SDLA-SRA-3	Threat Modeling	A threat model shall be created and documented for the component or the system.	Verify that a threat model exists for the component or system, and that it is documented.	Verify that abuse case testing is required as part of the development process. Verify that an abuse case test plan was created for any product or system that has been developed according to the same process being evaluated.	No	Microsoft Stage 3: Product Risk Assessment ISO/IEC 15408-3: AVA_VAN.2.3E CLASP: Perform security analysis of system requirements and design (threat modeling)	1, 2, 3, 4	
X	X	SDLA-SRA-3.1	CM of Threat Models	The threat model documents shall be placed under configuration management	Verify that the threat model document is are in the CM system.	Verify that the development process requires that the threat model is placed under configuration management. Or verify that the threat model for any component or system developed according to the same process being evaluated has been placed under configuration management.	No	Microsoft Stage 4: Risk Analysis	1, 2, 3, 4	
X	X	SDLA-SRA-3.2	Threat Model Updates	The threat model shall be updated whenever the design changes unless the changes do not affect the threat model.	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.	No	Microsoft Stage 4: Risk Analysis	1, 2, 3, 4	
X	X	SDLA-SRA-3.3	Threat Model Inclusion	All subsystems within the trust boundary of the component or system, shall be included in the threat model	Inspect the threat model and verify that all subsystems within the trust boundary have been included in the threat model.	Verify that the development process requires that all subsystems within the trust boundary 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 all subsystems within the trust boundary.	No	Microsoft Stage 4: Risk Analysis	1, 2, 3, 4	
X	X	SDLA-SRA-3.4	Use and Misuse Scenarios	The threat model shall define both use and mis-use scenarios	Inspect the threat model and verify that both use and mis-use scenarios are included.	May verify that the development process requires use and misuse scenarios to be included in the threat model. Or may verify that the threat model for any component or system developed according to the same process being evaluated includes both use and misuse scenarios.	No	Microsoft Stage 4: Risk Analysis CLASP: Detail misuse cases	1, 2, 3, 4	
X	X	SDLA-SRA-3.5	External Dependencies	The threat model shall include a list of external dependencies	Inspect the threat model and verify that external dependencies are listed or that it explicitly states that there are none.	Verify that the 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.	No	Microsoft Stage 4: Risk Analysis CLASP: Document Security-Relevant Requirements	1, 2, 3, 4	
X	X	SDLA-SRA-3.6	External Security Notes	The threat model shall include external security notes to describe the security boundaries, and document how administrators and application designers can maintain security when using the component or system.	Inspect the threat model and verify that external security notes are included.	May verify that the development process requires that external security notes are included in the threat model. Or may verify that the threat model for any component or system developed according to the same process being evaluated includes external security notes.	No	Microsoft Stage 4: Risk Analysis	1, 2, 3, 4	
X	X	SDLA-SRA-3.7	Data Flow Diagrams	The threat model shall include or reference data flow diagrams or an equivalent method of modeling system behavior	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 development process requires that data flow diagrams or equivalent method 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 data flow diagrams or an equivalent method.	No	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling)	1, 2, 3, 4	
X	X	SDLA-SRA-3.8	Trust Boundaries	Trust boundaries shall be included in the data flow diagrams/system behavioral model	Verify that trust boundaries are documented in the data flow diagram or equivalent system behavioral model.	May verify that the development process requires that trust boundaries are included in the threat model. Or may verify that the threat model for any component or system developed according to the same process being evaluated includes trust boundaries.	No	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling)	1, 2, 3, 4	
X	X	SDLA-SRA-3.9	Threats	The threat model shall document threats to the component or system.	Verify that threat model documents a list of threats and that the list includes, at a minimum, the threats identified in SDLA-SRS-3.2	Verify that the 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.	No	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling)	1, 2, 3, 4	
X	X	SDLA-SRA-3.10	Risk Levels	Threats shall all be assigned risk levels	Verify that each threat is defined a risk level, and that the risk levels are well defined	Verify that the development process requires that each threat in the threat model is assigned a risk level, and that the risk levels are clearly defined.	No	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling)	1, 2, 3, 4	This specification does not prescribe a specific risk level scale. However, applicants must establish a scoring system and are encouraged to adopt a standardized scoring system such as the Common Vulnerability Scoring System (CVSS)
X	X	SDLA-SRA-3.11	Threat Mitigation	All threats above some defined risk level must be mitigated either by changing the component or system, or by requiring compensating controls at the time of integration.	Verify that all threats above the defined risk level have documented mitigations.	Verify that a procedure exists stating that all threats above a defined risk level must be mitigated. Verify that the defined risk level is defined, and covers a majority of the risk levels.	No	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling) ISO/IEC 15408-3: AVA_VAN.1.4E	1, 2, 3, 4	This specification does not prescribe a specific risk threshold above which all vulnerabilities must be mitigated. However, applicants are must establish a threshold based upon their risk scoring system (see SRA-3.11) above which all vulnerabilities must be mitigated. Applicants are encouraged to adopt a standardized scoring system such as the CVSS and to establish a risk threshold score. For example, CVSS scores above 7 are considered "High" and must be mitigated.

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-DSD-1	Modular Design	For each major component/subsystem in the description of the component or system architecture design, further refinement of the design shall be based on a partitioning into software modules or security zones which shall be documented in the detailed component or system design description. The design of each software module or security zone shall be specified including the purpose, interface, parameters, and effects of each module on the security functions.	Inspect detailed component or system design specification and verify that the design is broken down into modules. Also, verify that the module design is specified including purpose, interface, parameters and effects of the modules on security functions	Verify that the component or system development process requires that the design is broken down into modules which are documented in detailed design specifications. Verify that this was done for any project using the same development process that is under evaluation.	No	IEC 61508-3: 7.4.5.3 & 7.4.5.4 ISO/IEC 15408-3: ADV_INT.1.1D, ADV_INT.1.1C, ADV_INT.1.2C, ADV_TDS.3.2C, ADV_TDS.3.6C, & ADV_TDS.3.7C DO-178B: 11.10C	1, 2, 3, 4	
X	X	SDLA-DSD-1.1	Module or Zone Interfaces	The detailed component software or system design shall describe the purpose and method of use of all interfaces, providing details of expected input/output criteria, effects, exceptions and error messages, as appropriate.	Inspect the detailed component or system design and verify that relevant details of the module's provided interfaces are included	Verify that the component or system development process requires that the design must describe the purpose and method of use all interfaces to modules providing details of effects, exceptions and error messages as appropriate. Or verify that this was done for any project using the same development process that is under evaluation.	No	ISO/IEC 15408-3: ADV_TDS.3.8C	3, 4	
X	X	SDLA-DSD-1.2	Independent Modules or Zones	The detailed component or system design description shall describe how the design provides for largely independent modules or zones that avoid unnecessary interactions	Inspect detailed component or system design specification and verify that a description of how the design provides for largely independent modules or zones is included and is clear and logical.	May verify that component or system development process includes a checklist or some guidelines for design best practices which include having largely independent modules or zones that avoid unnecessary interfaces. Or may inspect detailed component or system design specification for any component or system developed according to the process under evaluation and verify that a description of how the design provides for largely independent modules or zones is included and is clear and logical.	No	ISO/IEC 15408-3: ADV_INT.1.1D & ADV_INT.1.3C	3, 4	
X	X	SDLA-DSD-1.3	Purpose and relationship between modules or zones	The design shall describe its purpose and relationship between modules and zones.	Inspect the detailed component or system design specification and verify that the interrelationships between modules or zones are documented.	May verify that the component or system development process requires that the component or system design must describe the interrelationships between modules or zones. Or may verify that this was done for any project using the same development process that is under evaluation.	No	ISO/IEC 15408-3:2008, ADV_TDS.3.7C	3, 4	
X	X	SDLA-DSD-1.4	Security Functions	The detailed component or system design shall describe how each security policy enforcing function is provided	Inspect the detailed component or system design and verify that the design of each security function is provided.	Verify that the component or system development process states that the detailed component or system design must describe how each security function is provided.	No	ISO/IEC 15408-3: ADV_TDS.3.2C & ADV_TDS.3.6C	3, 4	
X	X	SDLA-DSD-1.5	Externally Visible Interfaces	The detailed component or system design shall identify which of the interfaces to the modules are externally visible.	Inspect the detailed component or system design and verify that it identifies which of the interfaces to the modules are externally visible.	May verify that the software development process requires that the software design must identify which of the interfaces to the modules are externally visible. Or may verify that this was done for any project using the same development process that is under evaluation.	No	ISO/IEC 15408-3: ADV_TDS.2.8C & ADV_TDS.3.10C	3, 4	
X	X	SDLA-DSD-2	Secure Design Best Practice	The detailed component or system design process shall incorporate secure design best practices. This applies to all features, not just security features.	Verify that the development process has been followed in the area of secure best practices. Verify that some of the secure best practices defined in this requirement have been employed in the development of the component or system being evaluated.	Verify that secure best practices are documented as part of the process, and that some mechanism is in place to ensure that they are followed (for example a review with a checklist).	No	CLASP: Apply security principals to design	1, 2, 3, 4	Typical best practices include economy of mechanism, fail-safe defaults, complete mediation, open design, separation of privilege, least privilege, least common mechanism, psychological acceptability. When considering off-the-shelf technologies, perform a risk assessment of the technology before designing it into the system. At least some of these practices should be included on the list of best practices.
X	X	SDLA-DSD-3	Input Validation	Input validation shall be performed wherever data can enter the system or cross a trust boundary. Validation should check for both the receipt of inputs when they are not expected when in a given state, and unexpected values of fields in inputs that are expected.	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.	No	CLASP: Apply security principals to design & Implement Security Contracts	1, 2, 3, 4	
X	X	SDLA-DSD-4	Data security policy	The detailed component or system design should document the security policy for all data (i.e. which user roles or service roles can access the data)	May inspect the detailed component or system design specification and verify that it documents the security policy for specific data.	May verify that the software development process states that the detailed design should document the security policy for all data.	No	CLASP: Apply security principals to design	2, 3, 4	
X	X	SDLA-DSD-5	Time Sequencing	The detailed component or system design description should include scheduling procedures and inter-process/inter-task communications mechanisms	Inspect the detailed component or system design specification and verify that it documents all relevant time sequencing information	Verify that the software development process requires that the software design must describe scheduling procedures and inter-process/inter-task communications mechanisms. Or may verify that this was done for any project using the same development process that is under evaluation.	No	DO-178B: 11.10f	2, 3, 4	For example including rigid time sequencing , preemptive scheduling, and interrupts

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-DSG-1	User Documented Security Guidelines	User documentation shall include security guidance for administrators and administrators	Verify that existence of documented security guidelines for administrators and administrators (unless the product does not contain any administrator functionality)	Verify that the development process states that security guidelines for administrators and administrators must be included in user documentation. Verify that this was done for a product developed with the development process being evaluated.	No	IEC 61508-3: 7.6.2.1.b CLASP: Build user documented Security Guide ISO/IEC 15408-3: AGD_OPE.1.1C Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	1, 2, 3, 4	
X	X	SDLA-DSG-1.1	Actions and Constraints	Security Guidelines for administrators shall contain actions and constraints that are necessary to prevent security breaches	Inspect security guidelines for administrators and confirm that they contain actions and constraints related to security.	Inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm that they contain actions and constraints related to security. Or verify that the development process states that security guidelines for administrators must contain this information.	No	IEC 61508-3: 7.6.2.1.b	1, 2, 3, 4	
X	X	SDLA-DSG-1.1.1	Pre-installation Requirements	The security guidelines for administrators shall document any environmental requirements that must be satisfied before the component or system is installed, as required to meet typical end customer scenarios and security objectives.	Inspect the security guidelines for administrators and verify that they describe environmental requirements that must be satisfied before the component or system is installed. If not, determine if any such requirements are needed.	Inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm that they describe environmental requirements that must be satisfied before the component or system is installed. Or verify that the development process states that security guidelines for administrators must contain this information.	No	CLASP: Build user documented Security Guide ISO/IEC 15408-3: AGD_PRE.1.2C	1, 2, 3, 4	
X	X	SDLA-DSG-1.1.2	Installation Requirements	The security guidelines for administrators shall outline the best practices that should be adhered to when installing the product.	Inspect the security guidelines for administrators and verify that they describe best practices that should be adhered to when installing software.	Inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm that they outline the best practices that should be adhered to when installing the product. Or verify that the development process states that security guidelines for administrators must contain this information.	No	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	1, 2, 3, 4	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.
X	X	SDLA-DSG-1.1.2.1	Security Configuration Options	The security guidelines for administrators shall list, and explain all security configuration options present in the system, and make note of their default and recommended settings.	Inspect the security guidelines for administrators and verify that they describe all security configuration options including default and recommended settings.	Inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm 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.	No	CLASP: Build user documented Security Guide Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	1, 2, 3, 4	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.
X	X	SDLA-DSG-1.1.2.2	Secure installation by default	The installation shall install the product as secure by default so that the default configuration is considered secure without any additional configuration changes.	Verify that this requirement is documented as a product requirement and that validation testing was done to show that the requirement was met.	Requirement not applicable to the development process	No		1, 2, 3, 4	
X	X	SDLA-DSG-1.1.3	Secure Administration	The security guidelines for administrators shall include guidance that describes how to administer the product in a secure manner.	Inspect security guidelines for administrators and confirm 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 administrators for a product developed with the development process being evaluated and confirm 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 for administrators must contain this information.	No	ISO/IEC 15408-3: AGD_OPE.1.2C	1, 2, 3, 4	
X	X	SDLA-DSG-1.1.3.1	Administrator warnings	The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment	Inspect security guidelines for administrators and confirm that they contain warnings about functions and privileges that should be controlled in a secure processing environment (unless the product does not have administrative capability)	May inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm that they contain warnings about functions and privileges that should be controlled in a secure processing environment. Or may verify that the development process states that security guidelines for administrators must contain this information.	No	ISO/IEC 15408-3: AGD_OPE.1.3C	1, 2, 3, 4	
X	X	SDLA-DSG-1.1.3.2	Administrator Assumptions	The administrator guidance shall describe all assumptions regarding administrator behavior that are relevant to secure operation of the product	Inspect security guidelines for administrators and confirm that they contain assumptions regarding administrator behavior that are relevant to secure operation (unless the product does not have administrative capability)	May inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm that they describe all assumptions regarding administrator behavior that are relevant to secure operation of the product. Or may verify that the development process states that security guidelines for administrators must contain this information.	No	ISO/IEC 15408-3: 2005: AGD_ADM.1.4C	1, 2, 3, 4	User behavior is defined as actions that a user that does not have administrative privileges may take.
X	X	SDLA-DSG-1.1.4	Administrator Guidance	The security guidelines for administrators shall include administrator guidance that clearly presents all administrator responsibilities necessary for secure operation of the product, including those related to assumptions regarding administrator behavior found in the statement of product security environment	Inspect security guidelines for administrators and confirm that they include administrator responsibilities necessary for secure operation of the product. When applying this requirement to a system, verify that system level administrator documentation has been created to document administrator responsibility	May inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm that they include administrator guidance that clearly presents all administrator responsibilities necessary for secure operation of the product, including those related to assumptions regarding administrator behavior found in the statement of product security environment. Or verify that the development process states that security guidelines for administrators must contain this information.	No	ISO/IEC 15408-3: AGD_OPE.1.6C	1, 2, 3, 4	System level user documentation is usually created by the integrator.

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-DSG-1.1.5	Known Security Risks	The security guidelines for administrators shall document any known security risks that the customer can take action to mitigate, along with recommended compensating controls, such as recommended third party software that can mitigate the issue, firewall configurations, or intrusion detection signatures.	Verify that known security risks are included in security guidelines for administrators. Verify any administrator documented recommendations made during threat modeling, attack surface reduction or security design reviews have been included. 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 administrators for a product developed with the development process being evaluated and confirm that they include known security risks. Verify any administrator documented recommendations made during threat modeling, attack surface reduction or security design reviews have been included. 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 for administrators must contain this information.	No	CLASP: Build user documented Security Guide Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	1, 2, 3, 4	
X	X	SDLA-DSG-1.1.6	API Security	If an API (Application Programming Interface) or set of classes or objects that developers can use to build applications is provided, security information and best practices shall be provided for each applicable function or method call.	If the product contains an API or a set of classes or objects that developers can use, verify that security information and best practices are provided for each applicable function or method call.	May inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm that If the product contains an API or a set of classes or objects that developers can use then security information and best practices are provided for each applicable function or method call. Or may verify that the development process states that security guidelines for administrators must contain this information.	No	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	1, 2, 3, 4	
X	X	SDLA-DSG-1.2	Reporting Security Vulnerabilities	The security guidelines for users and administrators shall contain procedures for reporting security vulnerabilities back to the product manufacturer.	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.	No	IEC 61508-3: 7.6.2.1.f	1, 2, 3, 4	
X	X	SDLA-DSG-1.3	Security Architecture	The security guidelines for administrators shall document the security architecture including the threat profile assumed in design and the high-level security functionality of the system as relevant to the user — including authentication mechanisms, default policies for authentication and other functions, and any security protocols that are mandatory or optional.	Verify that the security architecture is included in the security guidelines for administrators including assumed threat profile, high-level security functionality, and security protocols.	May inspect security guidelines for administrators for a product developed with the development process being evaluated and confirm that they include describe the security architecture including assumed threat profile, high-level security functionality, and security protocols. Or may verify that the development process states that security guidelines for administrators must contain this information.	No	CLASP: Build user documented Security Guide	1, 2, 3, 4	
X	X	SDLA-DSG-1.3.1	Administrator Functions	The security guidelines for administrators shall document the functions and interfaces available to the administrator of the product.	Inspect the security guidelines for administrators and verify that administrator functions and interfaces are documented (unless the product has none).	May inspect security guidelines for administrators for a product developed with the development process being evaluated and verify that administrator functions and interfaces are documented (unless the product has none). Or may verify that the development process states that security guidelines for administrators must contain this information.	No	ISO/IEC 15408-3: AGD_OPE.1.1C	1, 2, 3, 4	
X	X	SDLA-DSG-1.3.2	User Functions	The security guidelines for users shall document the functions (including usage) and interfaces available to non-administrative administrators of the product.	Inspect the security guidelines for users and verify that user functions and interfaces are documented (unless the product has none). When applying this requirement to a system, verify that system level user documentation has been created to document user functions and interfaces created during system integration.	May inspect security guidelines for users for a product developed with the development process being evaluated and verify that user functions and interfaces are documented (unless the product has none). Or may verify that the development process states that security guidelines for administrators must contain this information.	No	ISO/IEC 15408-3: AGD_OPE.1.1C	1, 2, 3, 4	For systems, the user functions available may be limited to those configured by the integrator. So in this case the integrator must produce this documentation.
X	X	SDLA-DSG-2	Operation and Maintenance Instructions	Operation and Maintenance instructions, which document how to use the product correctly, shall be provided.	Verify the existence of operation and maintenance instructions.	Verify that the development process states that operation and maintenance instructions must be created for each product.	No	IEC 61508-3: 7.6.2.5	1, 2, 3, 4	
X	X	SDLA-DSG-3	User Manual Review	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	Verify that all user manuals were reviewed by security experts by reviewing meeting minutes and confirming 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	No	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	1, 2, 3, 4	
X	X	SDLA-DSG-4	Security Tools	If security tools to help administrators set a secure configuration and audit against a secure baseline have been created, then they should be documented in the security guidelines.	Determine if such tools exist, and if so verify that their usage is described in the security guidelines.	None required.	No	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	1, 2, 3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-MIV-1	Security Coding Standard	Software shall be developed compliant with a security coding standard	Confirm 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 confirm that the coding standard is being followed by reviewing artifacts such as code review minutes or static analysis results or by looking at code.	No	IEC 61508-7.4.4.5	2, 3, 4	The security coding standard does not have to be an independent document. It may, for example, be part of an overall coding standard.
X	X	SDLA-MIV-1.1	Source Code Documentation	The security coding standard shall specify procedures for source code documentation	None Required	Verify that the security coding standard includes procedures for source code documentation.	No	IEC 61508-7.4.4.6	1, 2, 3, 4	
X	X	SDLA-MIV-1.2	Potentially exploitable coding constructs	The security coding standard should include a list of potentially exploitable coding constructs or designs that should not be used. This list should be obtained from a recognized source and should be based on real world security attacks	None Required	Verify that the 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 CERT secure coding standards are a recommended source. If this source is not used, the coding standard should be comparable to the CERT secure coding standards.	No	IEC 61508-7.4.4.6 Microsoft: Stage 6: Secure Coding Policies	2, 3, 4	
X	X	SDLA-MIV-1.3	Banned Functions	The security coding standard shall include a list of functions that are banned because they have been deemed to cause a security risk and alternative functions are available that mitigate that risk.	None Required	Verify that the security coding standard includes banned functions.	No	Microsoft: Stage 6: Secure Coding Policies	1, 2, 3, 4	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().
X	X	SDLA-MIV-2	Source Code Review	Source code shall be reviewed to make sure that it is clear and understandable and to find security bugs. A security checklist should be used during the review.	Verify that some code has been reviewed, 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.	Verify that procedures state that code must be reviewed. Verify that a security checklist exists and must be used as part of the review. 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	No	IEC 61508-3: 7.4.6.1 & 7.4.6.2 CLASP: Perform Source Level Security Review Microsoft Stage 8: The Security Push	2, 3, 4	Source code review requirements (MIV 2, 2.1, 2.2) apply to all code developed by the applicant.
X	X	SDLA-MIV-2.1	Code Reviews - High ISASecure Levels	All code shall be reviewed unless documented evidence exists to show that a particular module can not contain any security vulnerabilities.	Verify that the list of code that has been reviewed includes all software modules except for ones that have documented evidence that a module cannot contain any security vulnerabilities.	Verify that procedures state that all code must be reviewed unless documented evidence exists to show that a particular module can not contain any security vulnerabilities.	No	IEC 61508-3: 7.4.6.1 & 7.4.6.2 CLASP: Perform Source Level Security Review Microsoft Stage 8: The Security Push	3, 4	Source code review requirement (MIV 2.1) also applies to open source code or third-party source code that has been integrated into the applicant's product or system. In the case of third-party compiled code (e.g. binary libraries) the applicant is responsible, as part of their software supply chain security risk management program, to ensure that the third-party supplier of the code has performed appropriate code reviews.
X	X	SDLA-MIV-2.2	Code Reviews - Medium ISASecure Levels	At a minimum, code that meets the following criteria shall be reviewed: -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	Verify that the list of code that has been reviewed includes all code which meets the stated criteria.	Verify that procedures state that all code which meets the stated criteria must be reviewed.	No	IEC 61508-3: 7.4.6.1 & 7.4.6.2 CLASP: Perform Source Level Security Review Microsoft Stage 8: The Security Push	2, 3, 4	Source code review requirement (MIV 2.2) apply to all code developed by the applicant.
X	X	SDLA-MIV-2.3	Software Module Size	During code reviews software module size shall be reviewed. Modules that are too long or complex to easily be understood and tested should be broken up into smaller modules.	Verify that the version of the checklist used during the code reviews includes reviewing the module size.	Verify that a code review checklist exists and that reviewing modules size is included in the checklist.	No	IEC 61508-3: 7.4.6 Table B.9	2, 3, 4	
X	X	SDLA-MIV-2.4	Justification of code privilege	During code reviews, all code running as Local System or with Admin privileges shall be reviewed to ensure that it has valid reasons for doing so.	Verify that the version of the checklist used during the code reviews includes reviewing whether the code is running at the correct privilege.	Verify that a code review checklist exists and reviewing whether the code is running at the correct privilege should be included in the checklist.	No		2, 3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-MIV-3	Static Analysis - High ISASecure Levels	A static security analysis tool shall be run on all source code, including third-party source code, to check the code for potential security problems.	Verify that security static analysis tools has been run on all source code and that the results have been documented.	Verify that the development procedures state that security static analysis tools should be run on all source code and that the results must be documented. 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.	No	CLASP: Perform Source Level Security Review Microsoft: Stage 6: Secure Coding Policies	3, 4	MIV 3 applies to all code developed by the applicant. It also applies to open source code or third-party source code that has been integrated into the applicant's product or system. In the case of third-party compiled code (e.g. binary libraries) the applicant is responsible, as part of their software supply chain security risk management program, to ensure that the third-party supplier of the code has performed appropriate static analysis.
X	X	SDLA-MIV-3.1	Static Analysis - Medium ISASecure Levels	A static security analysis tool shall be run on all source code, including third-party source code, that meets the following criteria: -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	Verify that static analysis has been run on all source code that meets the stated criteria and that the results have been documented.	Verify that the development procedures state that security static analysis tools should be run on all source code that meets the stated criteria and that the results must be documented. 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.	No	CLASP: Perform Source Level Security Review Microsoft: Stage 6: Secure Coding Policies	2, 3, 4	MIV 3.1 applies to all code developed by the applicant. This also apply to open source code or third-party source code that has been integrated into the applicant's product or system. In the case of third-party compiled code (e.g. binary libraries) the applicant may perform static binary analysis OR ensure that the third-party supplier of the code has performed appropriate static analysis.
X	X	SDLA-MIV-3.2	Static Analysis Checks	The static analysis tool shall check for most of the potentially exploitable coding constructs defined in the security coding standard.	None Required	Verify that evidence exists showing that most of the potentially exploitable coding constructs 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.	No	CLASP: Perform Source Level Security Review Microsoft: Stage 6: Secure Coding Policies	2, 3, 4	
X	X	SDLA-MIV-3.3	Risk Mitigation	All risks identified by the static analysis tool in violation of the coding standard shall be mitigated unless the risk can be shown to be not relevant for one of the following reasons: • The risk is mitigated by an existing or recommended compensating control that is not within the scope of analysis for the tool. • The risk is not in the threat profile for the program. For example, attacks that require local user access to the same machine running the software may have already been deemed outside the scope of consideration. • The risk is a false positive in the analysis itself.	Verify that all risks identified by the static analysis tool have been either corrected or the reason for them not being relevant has been documented. If many items have been marked as not relevant, review a few of them and determine if the reasons given are sufficient.	Verify that the development procedure states that all risks identified by the static analysis tool in violation of the coding standard shall be mitigated unless the risk can be shown to be not relevant for one of the following reasons: • The risk is mitigated by an existing or recommended compensating control that is not within the scope of analysis for the tool. • The risk is not in the threat profile for the program. For example, attacks that require local user access to the same machine running the software may have already been deemed outside the scope of consideration. • The risk is a false positive in the analysis itself.	No	CLASP: Perform Source Level Security Review Microsoft: Stage 6: Secure Coding Policies	2, 3, 4	Security vulnerabilities discoed during static analysis should be provided to the supplier of the source code
X	X	SDLA-MIV-3.4	Automated Static Analysis	Static security analysis tools shall be automated so that potential security problems are identified as code is checked in to source code repository	None Required	Verify that static security analysis has been automated either by demonstration of the process or review of source management procedures or both.	No	CLASP: Perform Source Level Security Review	3, 4	
X	X	SDLA-MIV-4	Module/Unit Testing	Module/Unit testing shall be performed on all code, including third-party source code and binaries, that meets the following criteria: -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	Verify that documented evidence exists that module/unit testing was completed on all code that meets the stated criteria.	Verify that the development process states that all code that meets the stated criteria is module tested and that the results are documented. Pick a product that was developed with the same development process being evaluated and confirm that documented evidence exists that module/unit testing was completed on some code.	No	IEC 61508-3: 7.4.7	N/A (see child requirements) ~~3, 4~~	Module/unit testing requirements (MIV 4 - 4.4) apply to all code developed by the applicant. They also apply to open source code or third-party source code that has been integrated into the applicant's product or system. In the case of third-party compiled code (e.g. binary libraries) the applicant is responsible, as part of their software supply chain security risk management program, to ensure that the third-party supplier of the code has performed appropriate module/unit testing.
X	X	SDLA-MIV-4.1	Equivalence Classes	Module/Unit tests shall use equivalence classes and input partition testing to determine a suitable set of inputs to test.	Choose a few module test results to review and confirm that they used these concepts.	Verify that development process states that module/unit tests shall use equivalence classes and input partition testing to determine a suitable set of inputs to test.	No	IEC 61508-3: 7.4.7 and Table B.2	N/A 3, 4	Equivalence classes are used to come up with a small subset of all possible inputs with the highest possibility of finding the most errors. This is done by partitioning “the input domain of a program into a finite number of equivalence classes such that one can reasonably presume that a test of a representative value of each class is equivalent to a test of any other value

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-MIV-4.2	Boundary Value Analysis	Module/Unit tests shall use boundary value analysis to determine additional input values to test.	Choose a few module/unit test results to review and confirm that they used this concept.	Verify that development process states that module/unit tests shall use boundary value analysis to determine a additional inputs to test.	No	IEC 61508-3: 7.4.7 and Table B.2	3, 4	Boundary value analysis extends the equivalence class technique. The difference lies in how the values to be tested are chosen. Rather than choose any value within the equivalence class, you choose 1 or more values such that the edge of the equivalence class is the subject of the test. Explicitly test min, max, min minus one and max plus one (when integer).
X	X	SDLA-MIV-4.3	Code Coverage	Module/Unit tests shall ensure that input data is chosen so that at least 90% of all statements and branches are tested.	Choose a few module/unit test results to review and confirm that they used this concept.	Verify that development process states that module/unit tests shall test at least 90% of all statements and branches.	No	IEC 61508-3: 7.4.7 and Table B.2	3, 4	Both sides of each branch must be tested.
X	X	SDLA-MIV-4.4	Module Test Documentation	Module/Unit test results shall be documented. The documentation shall include the following: -Module under test -Date of test -Name of tester -Input Values Tested -Output Values Received -Code coverage achieved -Pass/Fail -List of any discrepancies found	Choose a few module/unit test results to review and confirm that all of the required information has been documented.	Verify that the development process states that module/unit test results shall be documented and that the stated information is included in that documentation. Pick a project developed with the same development process being evaluated and choose a few module/unit test results to review and confirm that all of the required information has been documented.	No	IEC 61508-3: 7.4.7	3, 4	
	X	SDLA-MIV-5	COTS Operating Systems	If the product includes a Commercial off the Shelf (COTS) operating system, then the operating system shall either meet the requirements of this development phase or be certified to Common Criteria EAL 3 or higher or be certified to a comparable security standard, or compensating controls must be included in the product to ensure that security vulnerabilities in the operating system do not result in vulnerabilities above a certain severity level in the product.	Verify if a commercial operating system is used. If so, confirm that a certificate exists from a qualified 3rd party to show that the operating system meets the ISCI criteria for software implementation and module verification or Common Criteria EAL 3 or higher or compensating controls in the product have been documented. If neither of these requirements are met, further analysis of the operating system development process is required If compensating controls are documented, verify that potential vulnerabilities of the operating system have been documented in the threat model.	Not applicable.	No		2, 3, 4	
X		SDLA-MIV-6	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.	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.	No		1, 2, 3, 4	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	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SIT-1	Fuzz Testing	Fuzz Testing shall be performed on all parsers that process data originating external to the security zone or component. Further, the supplier shall consider the threat model to identify other parsers that should be fuzz tested based on risk.	See Child Requirements	See Child Requirements	Partially. Fuzz testing of core protocols is conducted as part of Communications Robustness Testing (see ISASecure EDSA-310)	Microsoft: Stage 7: Secure testing policies	1, 2, 3, 4	Example parsers include configuration parsers which parse the controllers configuration, network protocol parsers which parse messages received via network protocols such as TCP/IP, UDP, etc, and API's (Application Program Interfaces) that allow other devices to integrate with the controller.
X	X	SDLA-SIT-1.1	Fuzz Test Plan	A Fuzz Test Plan shall be created documenting the fuzz testing that will be done. The plan shall include a list of all parsers that will be fuzzed, a description of how the fuzzing will be done, whether smart fuzzing or dumb fuzzing will be done, and the pass/fail criteria for the tests.	Verify that a fuzz test plan exists, and includes all of the information documented in the requirement. Also verify that fuzz test plan covers all parsers that parse data sent to the component or system.	Verify that the development process states that a fuzz test plan must be created and must include fuzz testing of all parsers that parse external data sent to the controller. 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	No	Microsoft: Stage 7: Secure testing policies	1, 2, 3, 4	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.
X	X	SDLA-SIT-1.2	Automatically Generated Test Cases	The files or packets that will be "fuzzed" shall be automatically generated so that a large number of test case (in the thousands) can be executed.	Review Fuzz test results and confirm that a large number of test cases were executed.	Verify that the development process states that the files or packets that will be "fuzzed" shall be automatically generated so that a large number of test case (in the thousands) can be executed. Or pick a product that is developed using the process under evaluation and review Fuzz test results and confirm that a large number of test cases were executed.	No	Microsoft: Stage 7: Secure testing policies	1, 2, 3, 4	Automated tools are commercially available for certain types of fuzz testing.
X	X	SDLA-SIT-1.3	Fuzz Test Results	Fuzz Test Results shall be documented. Test results shall include the date the tests were run, the name of the tester, the version of software for the device under test, and the results of each test including whether the test passed or failed, any discrepancies between the expected and actual results, and a reference to any problem reports written up based on the test.	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 test states that fuzz test results must be documented. Pick a product that is developed using the process under evaluation and confirm that fuzz test results were documented for that product.	No	Microsoft: Stage 7: Secure testing policies	1, 2, 3, 4	
X	X	SDLA-SIT-2	Abuse Case Testing	Abuse case testing shall be performed on the component or system to find vulnerabilities	See Child Requirements	See Child Requirements	Partially. General abuse case testing is conducted as part of Communications Robustness Testing (see ISASecure EDSA-310) and Systems Robustness Testing (see ISASecure SSA-310)	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	1, 2, 3, 4	
X	X	SDLA-SIT-2.1	Threat exploitation	Abuse case testing shall attempt to exploit all threats identified in the threat model that have been mitigated	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.	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.	No	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	1, 2, 3, 4	
X	X	SDLA-SIT-2.2	Abuse Case Test Plan	The abuse case tests shall be documented in a test plan. The plan shall include a list of test cases. For each test case the plan shall include a test objective, test procedure, and expected results.	Verify that an abuse case test plan exists and includes all of the items described in the requirement.	Verify that the development process states that an abuse case test plan shall be created. Pick a product that is developed using the process under evaluation and confirm that an abuse case test plan was created.	No	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	1, 2, 3, 4	
X	X	SDLA-SIT-2.3	Abuse Case Test Results	The results of the abuse case tests shall be documented. Test results shall include the date the tests were run, the name of the tester, the version of software for the device under test, and the results of each test including whether the test passed or failed, any discrepancies between the expected and actual results, and a reference to any problem reports written up based on the test.	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 confirm that abuse case test results were documented.	No	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	1, 2, 3, 4	
X	X	SDLA-SIT-3	Known Vulnerability Detection	Known vulnerability detection shall be performed on the component or system just prior release, and the results dated	See Child Requirements	See Child Requirements	Components: No Systems: Yes, Vulnerability Identification Testing (VIT) is conducted as part of Systems Robustness Testing (see ISASecure SSA-310)	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	1, 2, 3, 4	
X	X	SDLA-SIT-3.1	Known Vulnerability Detection Test Plan	A known vulnerability detection test plan shall be created documenting the tools used to perform the testing and their configuration. The plan shall also include a list of all components and interfaces to be tested, a description of how the testing will be performed, and the pass/fail criteria for the tests.	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 confirm that a known vulnerability detection test plan was created.	No	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	1, 2, 3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SIT-3.2	Known Vulnerability Detection Test Results	The results of the known vulnerability detection tests shall be documented. Test results shall include the date the tests were run, the name of the tester, the version of software for the component under test, the manufacturer and version of the vulnerability detection test tool, and the results of each test including whether the test passed or failed, any discrepancies between the expected and actual results, and a reference to any problem reports written up based on the test.	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 confirm that known vulnerability detection test results were documented.	No	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	1, 2, 3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SPV-1	Security Process Assessment	One or more persons shall be appointed to carry out a security process assessment in order to arrive at a judgment of the security achieved by the component or system	See child requirements	Verify that the development process states that one or more persons shall be appointed to carry out a security assessment in order to arrive at a judgment of the security achieved by the component or system.	No	IEC 61508-1: 8.2.1 Microsoft: Stage 9: The final security review DO-178B: 8.3	1, 2, 3, 4	
X	X	SDLA-SPV-1.1	Application to security lifecycle	The security process assessment shall be applied to all phases throughout the overall development lifecycle. Those carrying out the security process assessment shall consider the activities carried out and the outputs obtained during each phase of the development lifecycle and judge the extent to which the objectives and requirements of the company's security development lifecycle have been met for a given project.	Verify, by reviewing the security process assessment plan, that all phases of the security lifecycle are considered.	Not applicable.	No	IEC 61508-1: 8.2.3	1, 2, 3, 4	
X	X	SDLA-SPV-1.2	Assessment timing	The security process assessment shall be carried out throughout the development lifecycle and may be carried out after each lifecycle phase or after a number of lifecycle phases, subject to the overriding requirement that a security assessment shall be undertaken prior to application for certification.	Verify that a security process assessment was done prior to the product release.	Not applicable.	No	IEC 61508-1: 8.2.4	1, 2, 3, 4	
X	X	SDLA-SPV-1.3	Assessment Plan	A security process assessment plan shall be created unless security process assessment is part of a standard development process assessment. The plan shall include the following information: - Those who will perform the assessment - A description of the work that will be done in the assessment - The outputs from each assessment - The scope of the assessment - Resources Required - Level of independence of those undertaking the assessment - The competence of those undertaking the assessment	Verify that a security process assessment plan exists and contains the information documented in this requirement.	Not applicable.	No	IEC 61508-1: 8.2.8	1, 2, 3, 4	
X	X	SDLA-SPV-1.4	Assessment Results	At the conclusion of the security process assessment, the assessment results shall be documented including recommendations for acceptance, qualified acceptance or rejection.	Verify that assessment results were documented and was ultimately accepted. If a qualified acceptance was given, verify that any qualifying items have been addressed.	Not applicable.	No	IEC 61508-1: 8.2.10	1, 2, 3, 4	
X	X	SDLA-SPV-1.5	Competence of Assessors	Those carrying out the security process assessment shall be competent for the activities to be undertaken	Verify that development organization has documented justification of the competence of those who carried out the assessment. This evidence can be based on experience, education, training, and/or certifications.	Not applicable.	No		1, 2, 3, 4	
X	X	SDLA-SPV-1.6	Independence of Assessors	Those carrying out the security process assessment shall not be members of the team that developed the component or system	Verify that development organization has documented justification of the competence of those who carried out the assessment. This evidence can be based on experience, education, training, and/or certifications.	Not applicable.	No		1, 2, 3, 4	
X	X	SDLA-SPV-1.7	Threat Model Review	Just prior to releasing a product or a major product release, the threat model should undergo a review to confirm that the model is accurate, up to date and that the appropriate mitigations are in place.	May verify that the threat model review was done by confirming evidence of the meeting, such as meeting minutes, exists.	May verify that the development process states that the threat model should undergo a review just prior to releasing the product.	No	Microsoft: Stage 9: The final security review	1, 2, 3, 4	
X	X	SDLA-SPV-1.8	Security Bug Severity	All security bugs found should be logged in the bug tracking system with a severity or criticality assigned.	May verify that security bugs are identified as such in the bug tracking system and that a sampling of these bugs shows that they all have a severity or criticality assigned.	May verify that the development process states that all security bugs found should be logged in the bug tracking system with a severity or criticality assigned.	No	Microsoft: Stage 9: The final security review		
X	X	SDLA-SPV-1.9	Unfixed Security Bugs Review	A list of unfixed security bugs shall be available. This list shall be reviewed prior to application for certification by the security assessor(s) to confirm that no bugs have been "mistakenly" left unfixed. All unfixed bugs must either be below the specified threshold for severity or criticality, or be approved as an exception according to the appropriate approval procedure.	Verify that a list of unfixed security bugs exists and that there is evidence that it was reviewed prior to release. This evidence could be in the form of meeting minutes or in the form of a field in the bug tracking database that is updated when the review is done.	Verify that the development process states that a list of unfixed security bugs shall be available and that all unfixed bugs must either be below the specified threshold severity or criticality, or be approved as an exception according to the appropriate approval procedure.	No	Microsoft: Stage 9: The final security review	1, 2, 3, 4	Known vulnerabilities in 3rd party components must be included in this review.

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SRP-1	Vulnerability Reporting	A published mechanism shall exist for security vulnerabilities to be reported by external entities such as customers or security researchers.	Not Applicable.	Verify that the mechanism is made publically available, for example on the company's web site.	No	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	1, 2, 3, 4	Examples include a dedicated e-mail address or phone number to report potential security vulnerabilities.
X	X	SDLA-SRP-2	Vulnerability Response	A documented process shall exist for responding to all reported security vulnerabilities.	Not Applicable.	Verify that process exists and see child requirements.	No	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	1, 2, 3, 4	
X	X	SDLA-SRP-2.1	Vulnerability Analysis	All reported security vulnerabilities must be analyzed to determine if they are valid and whether they are a duplicate of a known vulnerability.	Not Applicable.	Verify that process includes this step.	No	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	1, 2, 3, 4	
X	X	SDLA-SRP-2.2	Vulnerability Bug Tracking	Reported Security vulnerabilities that are determined to be valid shall be logged in the bug tracking system with a severity or criticality assigned.	Not Applicable.	Verify that process includes this step, that a bug tracking system is in place, and that existing security vulnerabilities are assigned a severity or criticality.	No	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	1, 2, 3, 4	
X	X	SDLA-SRP-2.3	Vulnerability Remediation Plan	A plan for resolving each valid reported security vulnerability shall be established and followed.	Not Applicable.	Verify that the process includes this step.	No	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	1, 2, 3, 4	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	SDLA-SRP-2.4	Related Vulnerabilities	When fixing a reported vulnerability related vulnerabilities from the point of view of the attacker should be fixed as well.	Not Applicable.	Verify that the process includes this step.			1, 2, 3, 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.
X	X	SDLA-SRP-2.5	Vulnerability Modifications	At minimum, the standard modification process shall be followed when correcting any reported vulnerabilities.	Not Applicable.	Verify that process includes this step and that a standard modification process is documented.	No		1, 2, 3, 4	
X	X	SDLA-SRP-2.6	Root Cause Analysis	Root cause analysis shall be done for all reported security vulnerabilities	Not Applicable.	Verify that process includes this step and that root cause analysis has been done for existing vulnerabilities (ones that were found after this step became part of the process).	No	Microsoft: Stage 10: Security Response Planning	1, 2, 3, 4	
X	X	SDLA-SRP-2.7	Lessons Learned	Recommendations for changes that would prevent similar errors from occurring in the future shall be done for all vulnerabilities that are fixed.	Not Applicable.	Verify that process includes this step and that this was done for existing vulnerabilities (ones that were found after this step became part of the process).	No	Microsoft: Stage 10: Security Response Planning	1, 2, 3, 4	
X	X	SSDA-SRP-3	Modification Request	A modification shall be initiated only on the issue of an authorized software modification request	Not Applicable.	Verify that a process is in place to make an authorize a modification request. Verify that process states that all moderations must follow this process. Audit some recent modifications to see if they followed this process.	No	IEC 61508-3: 7.8.2.2	1, 2, 3, 4	
X	X	SSDA-SRP-4	Impact Analysis	An analysis shall be carried out on the impact of the proposed software modification on the security of the product or system; a) to determine whether or not the threat model shall be updated b) to determine which software security lifecycle phases will need to be repeated.	Not Applicable.	Verify that 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.	No	IEC 61508-3: 7.8.2.3 DO-178B: 7.2.5.b	1, 2, 3, 4	
X	X	SDLA-SRP-4.1	Verification and Validation	The impact analysis shall include a list of verification and validation tests and steps that will be executed for the change.	Not Applicable.	Verify that the audited impact analysis includes a list of verification and validation tests and steps that must be executed. Verify that these were executed. Verify that this step is called out for in the change process.	No	IEC 61508-3: 7.8.2.6	1, 2, 3, 4	
X	X	SDLA-SRP-4.2	Impact Analysis Documentation	The impact analysis shall be documented.	Not Applicable.	Covered by validation activity of parent.	No	IEC 61508-3: 7.8.2.4	1, 2, 3, 4	
X	X	SDLA-SRP-4.3	Return to appropriate phase	All modifications which have an impact on the security of the product shall initiate a return to an appropriate phase of the software security lifecycle. All subsequent phases shall then be carried out in accordance with the procedures specified for the specific phases.	Not Applicable.	Verify that this process is documented and evidence that it was followed can be found on audited changes.	No	IEC 61508-3: 7.8.2.5 DO-178B: 7.2.4.d	1, 2, 3, 4	
X	X	SDLA-SRP-4.3.1	Changes which impact security	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	Not Applicable.	Verify that this process is documented and evidence that it was followed can be found on audited changes.	No		1, 2, 3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SVT-1	Validation Planning	Planning shall be carried out to specify the steps, both procedural and technical, that will be used to demonstrate that the component or system satisfies its security requirements	Verify that a security validation test plan is created or that the general validation test plan has a section for security.	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 this was done for a product developed using the process under evaluation.	No	IEC 61508-3: 7.3.2.1 CLASP: Identify, Implement, and perform security tests	1, 2, 3, 4	
X	X	SDLA-SVT-1.1	Validation Planning Details	The plan for validating the component or system security shall consider the following: a) The goal of each test b) The techniques, procedures, and scenarios that shall be used for confirming that each security function conforms with the specified requirements for the software security functions c) specific reference to the specified requirements for software security; d) the required environment in which the validation activities are to take place (for example for tests this would include calibrated tools and equipment); e) pass/fail criteria for each test; f) The entirety of the test sets covers or satisfies all the security requirements for the device or system	Verify that the test plan includes the items listed in this requirement.	Verify that the development process or validation test plan template includes the items listed in this requirement.	No	IEC 61508-3: 7.3.2.2 ISO/IEC 15408-3: ATE_FUN.1.1C, ATE_FUN.1.2C, & ATE_FUN.1.3C	1, 2, 3, 4	
X	X	SDLA-SVT-1.1.1	Pass/Fail Criteria	The pass/fail criteria for accomplishing software validation shall include: a) the required input signals with their sequences and their values; b) the anticipated output signals with their sequences and their values and acceptable values; and c) other acceptance criteria, for example memory usage, timing and value tolerances.	Verify that the test plan includes the items listed in this requirement.	Verify that the development process or validation test plan template includes the items listed in this requirement.	No	IEC 61508-3: 7.3.2.5 IEC 154080-3: ATE_FUN.1.4C	1, 2, 3, 4	
X	X	SDLA-SVT-2	Validation Activities	The validation activities shall be carried out as specified during software security validation planning.	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.	No	IEC 61508-3: 7.7.2.2 CLASP: Identify, Implement, and perform security tests	1, 2, 3, 4	
X	X	SDLA-SVT-3	Validation Results	The results of software security validation shall be documented	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.	No	IEC 61508-3: 7.7.2.3 ISO/IEC 15408-3: ATE_FUN.1.1D, ATE_FUN.1.2D, ATE_FUN.1.1C	1, 2, 3, 4	
X	X	SDLA-SVT-3.1	Detailed Documentation	For each security function, software security validation shall document the following results: a) the version of the software security validation plan being used; b) the security function being validated (by test or analysis), along with reference to the software security validation plan; c) tools and equipment used; d) the results of the validation activity including pass/fail assessment; e) discrepancies between expected and actual results. f) references to any bug reports written up as a result of this test.	Verify that actual test results follow the validation test procedure or validation test results template.	Verify that either a template for validation testing exists or a procedure which documents what must be included in the test results. Verify that procedure or template includes the items from this requirement. Verify that actual test results follow this procedure or template for a product developed with the process under evaluation.	No	IEC 61508-3: 7.7.2.4 ISO/IEC 15408-3: ATE_COV.1.1D & ATE_COV.1.1C	1, 2, 3, 4	
X	X	SDLA-SVT-3.2	Discrepancies	When discrepancies occur between expected and actual results, the analysis made and the decisions taken on whether (1) to continue the validation, or (2) to issue a bug report and return to an earlier part of the development lifecycle, shall be documented as part of the results of the software security validation	Verify that test results procedure or template and actual test results include this information.	Verify that test results procedure or template and actual test results for a product developed with this process include this information.	No	IEC 61508-3: 7.7.2.5	1, 2, 3, 4	
X	X	SDLA-SVT-4	Validation Methods	The validation of security-related software shall meet the following requirements: a) testing shall be the main validation method for software; analysis may be used alone or in conjunction with testing for those requirements for which significant confidence cannot be obtained by testing alone b) the software shall be exercised by simulation of: — Normally expected inputs exercised using valid equivalence classes and boundary values, — anticipated occurrences, and — Unexpected inputs exercised using equivalence class selection of invalid values	Review product testing and verify that a majority of the requirements are validated by test rather than analysis or design alone.	Verify that validation test process states that software shall be exercised by simulation of inputs using valid and invalid equivalence classes.	No	IEC 61508-3: 7.7.2.6 DO-178B: 6.4.2.1 & 6.4.2.2	1, 2, 3, 4	

System	Component	Requirement ID	Requirement Name	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)	Validation by Independent Test Required (Yes/No)	Source of Requirement	*ISASecure* Level	Comments/Clarifications
X	X	SDLA-SRE-1	Concurrent Releases	Fixes for vulnerabilities above a given severity level should be released concurrently in all officially supported versions and languages of a product and to all customers, unless compensating controls can be put in place on existing releases to prevent the vulnerability from being exploited.	Not Applicable.	Verify that the process calls for concurrent releases of all versions when security vulnerabilities are fixed.	No	Microsoft: Stage 11: Product Release	1, 2, 3, 4	When a security patch is released, it is often reverse engineered by the hacker community so that previous versions can be exploited. Therefore all versions must be released as close as possible to each other so that the exploit can not be used on versions that do not yet have a patch.
X	X	SDLA-SRE-2	Watch for exploits	When vulnerabilities above a certain severity level are reported by external sources, but are not actively being exploited, the vendor shall actively watch for events that indicate that the vulnerability has been exploited or published.	Not Applicable.	Verify that the process calls for active watching of various sources to find out if a vulnerability is being exploited.	No	Microsoft: Stage 11: Product Release	1, 2, 3, 4	Security mailing lists or hacker web sites can be monitored. In addition error reports or intrusion detection logs from customers can be monitored.
X	X	SDLA-SRE-3	Report Vulnerabilities	When a vulnerability above a certain severity level has been published or has been exploited, customers should be notified of the vulnerability as well as any work-arounds that may exist to protect against the vulnerability.	Not Applicable.	Verify that the process documents customer notification for vulnerabilities above a certain severity level. If any such vulnerabilities exist, confirm that customer notification occurred for at least one of them	No		1, 2, 3, 4	
X	X	SDLA-SRE-4	Severe Vulnerabilities	If a severe vulnerability is being actively exploited then the following exceptions can be made to the process: -Releases for different versions or languages or customers can be released ahead of others -Related vulnerabilities can be released in a later release -The most obvious or critical vulnerabilities can be released first followed by a more complete update -The update could be released requiring manual installation with a more complete installation program to follow in a later update.	Not Applicable.	Verify that the development process still requires key parts of the standard process to be followed even in the case where a vulnerability is actively being exploited.	No	Microsoft: Stage 11: Product Release	1, 2, 3, 4	
X	X	SDLA-SRE-5	Patches	The development organization shall validate security patches from other vendors' products that are used in the development organization's own product (e.g. COTS OS, third-party source code or binaries). The validation must be timely (For example, within one week of release of the patch).	Not Applicable.	Verify that if the product being evaluated uses other products, then the development organization has a process in place to be notified when patches are available and to validate that they work properly with the product being evaluated.	No	ISCI Technical Committee	1, 2, 3, 4	