EDSA-312

ISA Security Compliance Institute - Embedded Device Security Assurance -
Software Development Security Assessment

Revision History

Version	Date	Changes
V1R3-03082010	2010.03.08	initial version published to http://www.ISASecure.org
1.4	2010.06.08	formatting changes

Number	Phase Name	Description
PH1	Security Management Process (SMP)	Process for planning an managing security development activities to ensure that security is designed into a product.
PH2	Security Requirements Specification (SRS)	Document customer driven security requirements, security features and the potential threats that drive the need for these features.
PH3	Software Architecture Design (SAD)	Top Level Software Design; Ensure that security is included in the design.
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)	Software Design down to the module level following security design best practices
PH6	Document Security Guidelines (DSG)	Create guidelines that users of the product must follow to ensure security requirements are met
PH7	Module Implementation & Verification (MIV)	Implement design by writing code following security coding guidelines. Ensure that software modules 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 and penetration testing
PH9	Security Process Verification (SPV)	Independent assessment that all required software development processes have been followed
PH10	Security Response Planning (SPR)	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.

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level[1]	Comments/Clarifications
Project Management					
SDSA-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.	IEC-61508-3: 6.2.1	All	
SDSA-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.	IEC-61508-1: 6.2.1b	All	
SDSA-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.	IEC-61508-1: 6.2.3, 6.2.4, & DO-178B: 4.2.g	All	
SDSA-SMP-1.3	Lifecycle Model	The development organization shall establish a life-cycle model to be used in the development and maintenance of the product. This model shall be documented.	IEC 61508-1: 6.2.1.c, DO-178B: 4.3 & ISO/IEC 15408-3: ALC_LCD.1.1D	All	
SDSA-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.	DO-178B: 4.3b	All	
SDSA-SMP-1.4	Basic Security Training	All people involved in software development of a product that has security concerns shall be given basic training in good security engineering practice and the secure development process that will be used on the project. In addition, software developers shall receive detailed training on common basic causes and mitigation techniques. 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.	CLASP: Institute security awareness program Microsoft: Stage 0: Education and awareness IEC 61508-1: 6.2.1.h	All	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.
SDSA-SMP-1.5	Competence	Those involved in software development of a product that has security concerns must be competent in carrying out the tasks assigned to them.	IEC 61508-1: 6.2.1.h	All	
SDSA-SMP-1.6	Development Tools	The development organization shall identify all development tools (including versions) used to create the product and document this information.	ISO/IEC 15408-3: ALC_TAT1.1D & IEC 61508-3: 7.4.4.2	All	NOTE: Unless otherwise notified the reference ISO/IEC 15408-3 hereafter designates the 2008 version, i.e. ISO/IEC 15408-3:2008.
SDSA-SMP-1.6.1	Development Tools Options	The development organization shall document the selected implementation-dependent options of the development tools in the security management plan.	ISO/IEC 15408-3: ALC_TAT1.2D	>2	
SDSA-SMP-1.7	Revision of security management plan	The software planning process should provide a means to revise the security management plan as a project progresses.	DO-178B: 4.2e	All	
SDSA-SMP-2	Action Item Resolution	A process shall exist for ensuring that action items from review meetings are documented and tracked to closure.	IEC-61508-1: 6.2.1.g	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level[1]	Comments/Clarifications
SDSA-SMP-3	Documentation of software releases	Software configuration management should formally document the release of security-related software.	IEC 61508-3: 6.2.3.f	All	
Development Environment Security					
SDSA-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 product design and implementation in its development environment.	ISO/IEC 15408-3: ALC_DVS.1.1.D & ALC_DVS.1.1.C	>1	
SDSA-SMP-4.1	Development Environment Security Evidence	The development environment security documentation shall provide evidence that these security measures are followed during the development and maintenance of the product.	ISO/IEC 15408-3: 2005: ALC_DVS.1.2C	>1	
Software Configuration Management					
SDSA-SMP-5	CM System	The development organization shall have a Configuration Management (CM) process.	ISO/IEC 15408-3: ALC_CMC.2.3C	All	
SDSA-SMP-5.1	Product Generation	The CM process shall provide an automated means to support the generation of the product.	ISO/IEC 15408-3: ACM_CMC.4.5C	>2	
SDSA-SMP-5.2	Ascertain Changes	The CM process shall provide an automated means to ascertain the changes between the product and its preceding version.	ISO/IEC 15408-3: ALC_CMC.5.9C	>2	
SDSA-SMP-5.4	Product Identification	The CM process shall provide a reference (unique identifier) for the product which shall be unique to each version of the product.	IEC 61508-3: 6.2.3.c & ISO/IEC 15408-3: ALC_CMC.1.1D & ALC_CMC.1.1C	All	
SDSA-SMP-5.4.1	Product Label	The product shall be labeled with its reference.	ISO/IEC 15408-3: ALC_CMC.1.1C	All	
SDSA-SMP-5.5	Authorized Changes	The CM process shall provide a means by which only authorized changes are made to the product implementation representation, and to all other configuration items.	ISO/IEC 15408-3: ALC_CMC.3.4C & IEC 61508-3: 6.2.3.d & 6.2.1.o	>1	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.
SDSA-SMP-5.6	Modification Audit	The CM process shall support the audit of all modifications to the product, including the originator, date, and time in the audit trail.	ISO/IEC 15408-3: ALC_CMC.5.9C & IEC 61508-3: 6.2.3.e	All	
SDSA-SMP-5.7	CM System Evidence	The CM shall document evidence that the CM system is operating in accordance with the CM plan.	ISO/IEC 15408-3: ALC_CMC.3.8C	>1	
SDSA-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.	ISO/IEC 15408-3: ALC_CMC.3.7C	>1	
SDSA-SMP-6	Configuration Management Plan	The development organization shall create a Configuration Management (CM) plan that defines how configuration items will be managed.	IEC 61508-3: 6.2.3.a & DO 178B: 4.3 & ISO/IEC 15408-3: ALC_CMC.3.5C	All	
SDSA-SMP-6.1	Automated CM Tools	The CM plan shall describe the automated tools used in the CM system.	ISO/IEC 15408-3: ALC_CMC.4.4C & ALC_CMC.4.5C	>2	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level[1]	Comments/Clarifications
SDSA-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.	ISO/IEC 15408-3: ALC_CMC.3.6C	>1	
SDSA-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.	IEC 61508-3: 6.2.1.o	All	
SDSA-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 product.	ISO/IEC 15408-3: 2005: ACM_CAP.4.13C & ACM_CAP.4.3C	>2	The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized
SDSA-SMP-7	Configuration List	The CM documentation shall include a configuration list of all configuration items that comprise the product and will be controlled by the CM process.	IEC 61508-3: 6.2.1.o & ISO/IEC 15408-3: ALC_CMC.1.1D	All	
SDSA-SMP-7.1	Configuration Item Description	The configuration list shall describe the configuration items that comprise the product.	ISO/IEC 15408-3: ALC_CMS.1.2C	All	
SDSA-SMP-7.2	Configuration Identification Method	The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the product.	ISO/IEC 15408-3: ALC_CMC.2.2C	All	
SDSA-SMP-7.3	CM System Identification	The CM process shall uniquely identify all configuration items that comprise the product.	ISO/IEC 15408-3: ALC_CMC.2.3C	All	
SDSA-SMP-7.4	Configuration Item Inclusion	The list of configuration items shall include all of the following items (see sub-requirements).		All	
SDSA-SMP-7.4.1	Configuration Item Inclusion	The list of configuration items shall include all items that make up the implementation representation of the product.	ISO/IEC 15408-3: ALC_CMS.3.1C	All	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.
SDSA-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.	ISO/IEC 15408-3: ALC_CMS.3.1C	All	
SDSA-SMP-7.4.3	Security Flaws	The list of configuration items shall include identified product security flaws.	ISO/IEC 15408-3: ALC_CMS.4.1C	>2	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.
SDSA-SMP-7.4.4	Development Tools	The list of configuration items shall include all development tools.	ISO/IEC 15408-3: ALC_CMS.5.1C	>2	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
General Requirements					
SDSA-SRS-1	Security Requirements Specification	A security requirements specification (SecRS) shall be created to document all required security functions of the product.	IEC 61508-3: 7.2.2.11, CLASP: Document security-relevant requirements ISO/IEC 14508-3: ASE_REQ.1.1D & ASE_OBJ.1.1D	All	The SecRS doesn't need to be single document. Many organizations create a security requirements section in other requirements and customer documents.
SDSA-SRS-2	Product Description	The developer shall provide a product description as part of the SecRS	ISO/IEC 14508-3: ASE_INT.1.1C	All	
SDSA-SRS-2.1	Scope of product	The product description shall describe the product or system type and the scope and boundaries of the product in general terms in both a physical and a logical way.	ISO/IEC 14508-3: ASE_INT_1.7C	All	
Operating Environment					
SDSA-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	CLASP: Specify operational environment, ISO/IEC 15408-3: ASE_SPD.1.4C	All	
SDSA-SRS-3.1	Operating Environment Assumptions	The statement of security environment shall identify and explain any assumptions about the intended usage of the product and the environment of usage that must be met by users in order for the product to be secure.	CLASP: Specify operational environment, ISO/IEC 15408-3: ASE_SPD.1.4C	All	These are the features provided by site security policies that are independent of the product. Examples: limited physical access, employee screening
SDSA-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 product or by its environment.	ISO/IEC 15408-3: ASE_SPD.1.1C	All	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					

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-SRS-4	Basic Security Functions	Required security functions/features that implement the required organizational security policies shall be included in the SecRS.	CLASP: Document security-relevant requirements	All	The source for these requirements could be the security business requirements or it could be based on standards such as ISA S99
SDSA-SRS-5	Security Assurance Level	Required security assurance level for the product should be included in the SecRS	IEC 61508: 7.2.2.11 ISO/IEC 15408-3: 2005: ASE_REQ.1.3C	All	Refer to ISA 99.01.01 for a definition of Security Assurance Level (SAL).
SDSA-SRS-6	Regulatory Requirements	Any regulatory requirements that the product must comply with should be included in the security requirements specification.		All	
Quality of Requirements					
SDSA-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.	IEC 61508-3: 7.2.2.3	All	
SDSA-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.	IEC 61508-3: 7.2.2.06.a, DO-178B: 5.1.2.e, f, & g, CLASP: Document security-relevant requirements	All	
SDSA-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.	IEC 61508-3: 7.2.2.4 & 7.2.2.06.c, DO-178B: 5.1.2.a ISO/IEC 14508-3: ASE_REQ.1.6C	All	
SDSA-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.	IEC 61508-3: 7.4.3.3	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-SAD-1	Product Partitioning	The software architecture design description shall be based on partitioning the product into components or subsystems	IEC 61508-3: 7.4.3.2.b DO-178B: 11.10.b & i ISO/IEC 15408-3: ADV_TDS.1.1D	All	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
SDSA-SAD-2	Network Design	The software architecture design shall describe the architecture of the system from the perspective of the network. This description shall include the actors that interact with the system as well as the system interconnection to related systems	CLASP: Identify Resources and Trust Boundaries	All	The design should show how subsystems are connected to each other on the network as well as how the product is networked to other related products.
SDSA-SAD-3	Data Resources	The software architecture design shall identify data resources that may be used or passed by the components of the system.	CLASP: Identify Resources and Trust Boundaries	All	Sample resources 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
SDSA-SAD-4	Trust Boundaries	The software architecture design shall document trust boundaries	CLASP: Identify Resources and Trust Boundaries Microsoft: Stage 4: Risk Analysis	All	Trust boundaries are demarcation points in the application that show where data moves from lower privilege to higher privilege
SDSA-SAD-5	Attack Surface	The software architecture design shall enumerate the attack surface which includes all possible entry points for an attacker.	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	All	Entry points include all interfaces, protocols and executing code. Interfaces include places where the file system is touched, local UI elements, inter-procedural communication points and any public methods that can be called externally when the program is running. Protocol documentation should include open network ports.
SDSA-SAD-6	Attack Surface Reduction	Attack surface reduction techniques shall be practiced to minimize the number of available entry points	Microsoft: Stage 2: Define and Follow Design Best Practices CLASP: Identify User Roles and Resource Capabilities	All	Entry points shall be minimized to only those absolutely necessary. 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.
SDSA-SAD-7	Semi-Formal Methods	The presentation of the software architecture shall be semi-formal.	ISO/IEC 15408-3: ADV_TDS.4.4C IEC 61508-3: 7.4.3	>2	Semi-formal is defined as expressed in a restricted syntax language with defined semantics. The language can be graphical or textual.

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-SAD-10	Interface Descriptions	The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the product, providing details of effects, exceptions and error messages, as appropriate.	ISO/IEC 15408-3: ACO_DEV.1.1C	>1	
SDSA-SAD-11	Secure Design Best Practice	The design process shall incorporate secure design best practices. This applies to all features, not just security features.	Microsoft: Stage 2: Define and Follow best design practices	All	
SDSA-SAD-12	Security Tools	Security tools to help users set a secure configuration and audit against a secure baseline shall be considered as part of the security design.	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	All	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.

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-SRA-1	Security Design Reviews	During the risk assessment, it shall be determined what portions of the project will require security design reviews	Microsoft Stage 3: Product Risk Assessment	All	
SDSA-SRA-2	Required Abuse Case Testing	During the risk assessment, it shall be determined what portions of the project will require abuse case testing	Microsoft Stage 3: Product Risk Assessment ISO/IEC 15408-3: AVA_VAN.1.3E	All	
SDSA-SRA-3	Threat Modeling	A threat model shall be created and documented for the product.	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)	All	
SDSA-SRA-3.1	CM of Threat Model	The threat model document shall be placed under configuration management	Microsoft Stage 4: Risk Analysis	All	
SDSA-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.	Microsoft Stage 4: Risk Analysis	All	
SDSA-SRA-3.3	Threat Model Periodic Updates	The threat model shall be updated at least once per year due to impact of ongoing security research	Microsoft Stage 4: Risk Analysis	All	
SDSA-SRA-3.4	Threat Model Inclusion	All subsystems within the trust boundary of the product shall be included in the threat model	Microsoft Stage 4: Risk Analysis	All	
SDSA-SRA-3.5	Use and Misuse Scenarios	The threat model shall define both use and mis-use scenarios	Microsoft Stage 4: Risk Analysis CLASP: Detail misuse cases	All	
SDSA-SRA-3.6	External Dependencies	The threat model shall include a list of external dependencies	Microsoft Stage 4: Risk Analysis CLASP: Document Security-Relevant Requirements	All	
SDSA-SRA-3.7	External Security Notes	The threat model shall include external security notes to describe the application's security boundaries and document how users and other application designers can maintain security when using the application.	Microsoft Stage 4: Risk Analysis	All	
SDSA-SRA-3.8	Data Flow Diagrams	The threat model shall include or reference data flow diagrams or an equivalent method of modeling system behavior	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling)	All	
SDSA-SRA-3.9	Trust Boundaries	Trust boundaries shall be included in the data flow diagrams/system behavioral model	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling)	All	
SDSA-SRA-3.10	Threats	The threat model shall document threats to product or system	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling)	All	
SDSA-SRA-3.11	Risk Levels	Threats shall all be assigned risk levels	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling)	All	
SDSA-SRA-3.12	Threat Mitigation	All threats above some defined risk level must be mitigated either by changing the product or by requiring compensating controls at the time of integration.	Microsoft Stage 4: Risk Analysis CLASP: Perform security analysis of system requirements and design (threat modeling) ISO/IEC 15408-3: AVA_VAN.1.4E	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-DSD-1	Modular Design	For each major component/subsystem in the description of the software architecture design, further refinement of the design shall be based on a partitioning into software modules which shall be documented in the detailed software design description. The design of each software module shall be specified including the purpose, interface, parameters, and effects of each module on the security functions.	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	All	
SDSA-DSD-1.1	Module Interfaces	The detailed software design shall describe the purpose and method of use of all interfaces to the modules, providing details of effects, exceptions and error messages, as appropriate.	ISO/IEC 15408-3: ADV_TDS.3.8C	>2	
SDSA-DSD-1.2	Independent Modules	The detailed software design description shall describe how the design provides for largely independent modules that avoid unnecessary interactions	ISO/IEC 15408-3: ADV_INT.1.1D & ADV_INT.1.3C	>2	
SDSA-DSD-1.4	Security Functions	The detailed software design shall describe how each security policy enforcing function is provided	ISO/IEC 15408-3: ADV_TDS.3.2C & ADV_TDS.3.6C	>2	
SDSA-DSD-1.5	Externally Visible Interfaces	The detailed software design shall identify which of the interfaces to the modules are externally visible.	ISO/IEC 15408-3: ADV_TDS.2.8C & ADV_TDS.3.10C	>2	
SDSA-DSD-2	Secure Design Best Practice	The detailed software design process shall incorporate secure design best practices. This applies to all features, not just security features.	CLASP: Apply security principals to design	All	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.
SDSA-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.	CLASP: Apply security principals to design & Implement Security Contracts	All	
SDSA-DSD-4	Data security policy	The detailed software design should document the security policy for all data (i.e. which user roles or service roles-can access the data)	CLASP: Apply security principals to design	>1	
SDSA-DSD-5	Time Sequencing	The detailed software design description should include scheduling procedures and inter-process/inter-task communications mechanisms	DO-178B: 11.10f	>1	For example including rigid time sequencing , preemptive scheduling, and interrupts

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-DSG-1	User Documented Security Guidelines	User documentation shall include security guidance for administrators and users	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	All	
SDSA-DSG-1.1	Actions and Constraints	Security Guidelines for Users shall contain actions and constraints that are necessary to prevent security breaches	IEC 61508-3: 7.6.2.1.b	All	
SDSA-DSG-1.1.1	Pre-installation Requirements	The security guidelines for users shall document any environmental requirements that must be satisfied before the system is installed, as required to meet typical end customer scenarios and security objectives.	CLASP: Build user documented Security Guide ISO/IEC 15408-3: AGD_PRE.1.2C	All	
SDSA-DSG-1.1.2	Installation Requirements	The security guidelines for users shall outline the best practices that should be adhered to when installing the product.	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	All	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.
SDSA-DSG-1.1.2.1	Security Configuration Options	The security guidelines for users shall list, and explain all security configuration options present in the system, and make note of their default and recommended settings.	CLASP: Build user documented Security Guide Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	All	
SDSA-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.		All	
SDSA-DSG-1.1.3	Secure Administration	The security guidelines for users shall include administrator guidance that describes how to administer the product in a secure manner.	ISO/IEC 15408-3: AGD_OPE.1.2C	All	
SDSA-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	ISO/IEC 15408-3: AGD_OPE.1.3C	All	
SDSA-DSG-1.1.3.2	Administrator Assumptions	The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the product	ISO/IEC 15408-3: 2005: AGD_ADM.1.4C	All	User behavior is defined as actions that a user that does not have administrative privileges may take.
SDSA-DSG-1.1.4	User Guidance	The security guidelines for users shall include user guidance that clearly presents all user responsibilities necessary for secure operation of the product, including those related to assumptions regarding user behavior found in the statement of product security environment	ISO/IEC 15408-3: AGD_OPE.1.6C	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-DSG-1.1.5	Known Security Risks	The security guidelines for users 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.	CLASP: Build user documented Security Guide Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	All	
SDSA-DSG-1.1.6	API Security	If an API 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.	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	All	
SDSA-DSG-1.2	Reporting Security Vulnerabilities	The security guidelines for users shall contain procedures for reporting security vulnerabilities back to the product manufacturer.	IEC 61508-3: 7.6.2.1.f	All	
SDSA-DSG-1.3	Security Architecture	The security guidelines for users 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.	CLASP: Build user documented Security Guide	All	
SDSA-DSG-1.3.1	Administrator Functions	The security guidelines for administrators shall document the functions and interfaces available to the administrator of the product.	ISO/IEC 15408-3: AGD_OPE.1.1C	All	
SDSA-DSG-1.3.2	User Functions	The security guidelines for users shall document the functions (including usage) and interfaces available to non-administrative users of the product.	ISO/IEC 15408-3: AGD_OPE.1.1C	All	
SDSA-DSG-2	Operation and Maintenance Instructions	Operation and Maintenance instructions, which document how to use the product correctly, shall be provided.	IEC 61508-3: 7.6.2.5	All	
SDSA-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	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	All	
SDSA-DSG-4	Security Tools	If security tools to help users set a secure configuration and audit against a secure baseline have been created, then they should be documented in the security guidelines.	Microsoft: Stage 5: Creating Security Documents, Tools, and Best Practices for Customers	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-MIV-1	Security Coding Standard	Software shall be developed compliant with a security coding standard	IEC 61508-7.4.4.5	>1	The security coding standard does not have to be an independent document. It may, for example, be part of an overall coding standard.
SDSA-MIV-1.1	Source Code Documentation	The security coding standard shall specify procedures for source code documentation	IEC 61508-7.4.4.6	All	
SDSA-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	IEC 61508-7.4.4.6 Microsoft: Stage 6: Secure Coding Policies	>1	
SDSA-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.	Microsoft: Stage 6: Secure Coding Policies	All	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().
SDSA-MIV-2	Code Review	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.	IEC 61508-3: 7.4.6.1 & 7.4.6.2 CLASP: Perform Source Level Security Review Microsoft Stage 8: The Security Push	>1	
SDSA-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.	IEC 61508-3: 7.4.6.1 & 7.4.6.2 CLASP: Perform Source Level Security Review Microsoft Stage 8: The Security Push	>2	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-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	IEC 61508-3: 7.4.6.1 & 7.4.6.2 CLASP: Perform Source Level Security Review Microsoft Stage 8: The Security Push	>1	
SDSA-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.	IEC 61508-3: 7.4.6 Table B.9	>1	
SDSA-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.		>1	
SDSA-MIV-3	Static Analysis - High ISASecure Levels	A static security analysis tool shall be run on all source code to check the code for potential security problems.	CLASP: Perform Source Level Security Review Microsoft: Stage 6: Secure Coding Policies	>2	
SDSA-MIV-3.1	Static Analysis - Medium ISASecure Levels	A static security analysis tool shall be run on all 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	CLASP: Perform Source Level Security Review Microsoft: Stage 6: Secure Coding Policies	>1	
SDSA-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.	CLASP: Perform Source Level Security Review Microsoft: Stage 6: Secure Coding Policies	>1	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-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.	CLASP: Perform Source Level Security Review Microsoft: Stage 6: Secure Coding Policies	>1	
SDSA-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	CLASP: Perform Source Level Security Review	>2	
SDSA-MIV-5	Module/Unit Testing	Module/Unit testing shall be performed on all 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	IEC 61508-3: 7.4.7	>2	
SDSA-MIV-5.1	Equivalence Classes	Module/Unit tests shall use equivalence classes and input partition testing to determine a suitable set of inputs to test.	IEC 61508-3: 7.4.7 and Table B.2	>2	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
SDSA-MIV-5.2	Boundary Value Analysis	Module/Unit tests shall use boundary value analysis to determine additional input values to test.	IEC 61508-3: 7.4.7 and Table B.2	>2	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).
SDSA-MIV-5.3	Code Coverage	Module/Unit tests shall ensure that input data is chosen so that at least 90% of all statements and branches are tested.	IEC 61508-3: 7.4.7 and Table B.2	>2	Both sides of each branch must be tested.

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-MIV-5.4	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	IEC 61508-3: 7.4.7	>2	
SDSA-MIV-6	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.		>1	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-SIT-1	Fuzz Testing	Fuzz Testing shall be performed on all parsers that parse external data sent to the controller.	Microsoft: Stage 7: Secure testing policies	All	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.
SDSA-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.	Microsoft: Stage 7: Secure testing policies	All	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.
SDSA-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.	Microsoft: Stage 7: Secure testing policies	All	Automated tools are commercially available for certain types of fuzz testing.
SDSA-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.	Microsoft: Stage 7: Secure testing policies	All	
SDSA-SIT-2	Product Abuse Case Testing	Abuse case testing shall be done on the product level to find vulnerabilities in the product.	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	All	
SDSA-SIT-2.1	Threat exploitation	Abuse case testing shall attempt to exploit all threats identified in the threat model that have been mitigated	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	All	
SDSA-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.	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	All	
SDSA-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.	Microsoft: Stage 7: Secure testing policies ISO/IEC 15408-3: AVA_VAN.1.3E CLASP: Identify, implement, and perform security tests	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-SPV-1	Security Assessment	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 product.	IEC 61508-1: 8.2.1 Microsoft: Stage 9: The final security review DO-178B: 8.3	All	
SDSA-SPV-1.1	Application to security lifecycle	The security assessment shall be applied to all phases throughout the overall development lifecycle. Those carrying out the security 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.	IEC 61508-1: 8.2.3	All	
SDSA-SPV-1.2	Assessment timing	The security 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.	IEC 61508-1: 8.2.4	All	
SDSA-SPV-1.3	Assessment Plan	A security assessment plan shall be created unless security 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	IEC 61508-1: 8.2.8	All	
SDSA-SPV-1.4	Assessment Results	At the conclusion of the security assessment, the assessment results shall be documented including recommendations for acceptance, qualified acceptance or rejection.	IEC 61508-1: 8.2.10	All	
SDSA-SPV-1.5	Competence of Assessors	Those carrying out the security assessment shall be competent for the activities to be undertaken		All	
SDSA-SPV-1.6	Independence of Assessors	Those carrying out the security assessment shall not be members of the team that developed the product.		All	
SDSA-SPV-1.7	Threat Model Review	Just prior to releasing a product, 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.	Microsoft: Stage 9: The final security review	All	
SDSA-SPV-1.8	Security Bug Severity	All security bugs found should be logged in the bug tracking system with a severity or criticality assigned.	Microsoft: Stage 9: The final security review		

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-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.	Microsoft: Stage 9: The final security review	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-SRP-1	Vulnerability Reporting	A published mechanism shall exist for security vulnerabilities to be reported by external entities such as customers or security researchers.	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	All	Examples include a dedicated e-mail address or phone number to report potential security vulnerabilities.
SDSA-SRP-2	Vulnerability Response	A documented process shall exist for responding to all reported security vulnerabilities.	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	All	
SDSA-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.	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	All	
SDSA-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.	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	All	
SDSA-SRP-2.3	Vulnerability Remediation Plan	A plan for responding to each valid reported security vulnerability shall be established and followed.	Microsoft: Stage 10: Security Response Planning CLASP: Build Vulnerability Remediation Procedures	All	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.
SDSA-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.			"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.
SDSA-SRP-2.5	Vulnerability Modifications	At minimum, the standard modification process shall be followed when correcting any reported vulnerabilities.		All	
SDSA-SRP-2.6	Root Cause Analysis	Root cause analysis shall be done for all reported security vulnerabilities	Microsoft: Stage 10: Security Response Planning	All	
SDSA-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.	Microsoft: Stage 10: Security Response Planning	All	
SSDA-SRP-3	Modification Request	A modification shall be initiated only on the issue of an authorized software modification request	IEC 61508-3: 7.8.2.2	All	
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.	IEC 61508-3: 7.8.2.3 DO-178B: 7.2.5.b	All	
SDSA-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.	IEC 61508-3: 7.8.2.6	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-SRP-4.2	Impact Analysis Documentation	The impact analysis shall be documented.	IEC 61508-3: 7.8.2.4	All	
SDSA-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.	IEC 61508-3: 7.8.2.5 DO-178B: 7.2.4.d	All	
SDSA-SRP-4.3.1	Changes which impact security	The following types of changes are considered to 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) 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 sets access controls or handles encryption keys or passwords		All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-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 software satisfies its security requirements	IEC 61508-3: 7.3.2.1 CLASP: Identify, Implement, and perform security tests	All	
SDSA-SVT-1.1	Validation Planning Details	The plan for validating the software 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	IEC 61508-3: 7.3.2.2 ISO/IEC 15408-3: ATE_FUN.1.1C, ATE_FUN.1.2C, & ATE_FUN.1.3C	All	
SDSA-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.	IEC 61508-3: 7.3.2.5 IEC 154080-3: ATE_FUN.1.4C	All	
SDSA-SVT-2	Validation Activities	The validation activities shall be carried out as specified during software security validation planning.	IEC 61508-3: 7.7.2.2 CLASP: Identify, Implement, and perform security tests	All	
SDSA-SVT-3	Validation Results	The results of software security validation shall be documented	IEC 61508-3: 7.7.2.3 ISO/IEC 15408-3: ATE_FUN.1.1D, ATE_FUN.1.2D, ATE_FUN.1.1C	All	
SDSA-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.	IEC 61508-3: 7.7.2.4 ISO/IEC 15408-3: ATE_COV.1.1D & ATE_COV.1.1C	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-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 safety validation	IEC 61508-3: 7.7.2.5	All	
SDSA-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	IEC 61508-3: 7.7.2.6 DO-178B: 6.4.2.1 & 6.4.2.2	All	

Requirement ID	Requirement Name	Requirement Description	Source of Requirement	*ISASecure* Level	Comments/Clarifications
SDSA-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.	Microsoft: Stage 11: Product Release	All	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.
SDSA-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.	Microsoft: Stage 11: Product Release	All	Security mailing lists or hacker web sites can be monitored. In addition error reports or intrusion detection logs from customers can be monitored.
SDSA-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.		All	
SDSA-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.	Microsoft: Stage 11: Product Release	All	
SDSA-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). The validation must be timely (For example, within one week of release of the patch).	ISCI Technical Committee	All	