ANSI/CAN/UL 2900-1:2023

JOINT CANADA-UNITED STATES
NATIONAL STANDARD

STANDARD FOR SAFETY

Software Cybersecurity for Network-Connectable Products, Part 1: General Requirements

Approved American National Standard
ANSI
ANSI/UL 2900-1-2023

SCC CCN

SCC FOREWORD

National Standard of Canada

A National Standard of Canada is a standard developed by a Standards Council of Canada (SCC) accredited Standards Development Organization, in compliance with requirements and guidance set out by SCC. More information on National Standards of Canada can be found www.scc.ca.

SCC is a Crown corporation within the portfolio of Innovation, Science and Economic Development (ISED) Canada. With the goal of enhancing Canada's economic competitiveness and social well-being, SCC leads and facilitates the development and use of national and international standards. SCC also coordinates Canadian participation in standards development, and identifies strategies to advance Canadian standardization efforts.

Accreditation services are provided by SCC to various customers, including product certifiers, testing laboratories, and standards development organizations. A list of SCC programs and accredited bodies is publicly available at www.scc.ca.

UL Standard for Safety for Software Cybersecurity for Network-Connectable Products, Part 1: General Requirements, ANSI/CAN/UL 2900-1

First Edition, Dated July 5, 2017

Summary of Topics

This revisions of ANSI/CAN/UL 2900-1, dated April 14, 2023, are issued to reflect the latest ANSI and SCC approval dates, and to include the following;

– Editorial Changes; 2.1, 6.7, 11.7, 15.5 and 15.6.

– Addition of Inclusive Language; 3.30, 8.5

– Clarification of Product Documentation; 4.1

– Updated Versions of Reference Material; 2.1, 6.1, 11.5

– Addition of Paragraph Numbering; 7.1.4 and 7.1.5

– Clarification of Definitions and Term Usage; 3.14A, 8.3 and 8.8

– Clarification of Sensitive Data Documentation; 10.1 and 15.1

– Removal of Redundant Statement; 11.5

– Self-Reference Correction; 12.3 and 12.5

– Clarification of Structured Penetration Testing Requirements Documentation; 16.1 and 16.2

– Clarification of Software Composition, Static Source Code Analysis and Static Binary and Bytecode Analysis Requirements Documentation; 3.42A, Section 13, 14.2, 17.1, 17.2, 17.3, Section 18, 19.2 – 19.5, Section A2, Figure A1.

Text that has been changed in any manner or impacted by ULSE's electronic publishing system is marked with a vertical line in the margin.

The new and revised requirements are substantially in accordance with Proposal(s) on this subject dated December 30, 2022.

Users of the electronic versions of UL's Standards for Safety agree to defend, indemnify, and hold ULSE harmless from and against any loss, expense, liability, damage, claim, or judgment (including reasonable attorney's fees) resulting from any error or deviation introduced while purchaser is storing an electronic Standard on the purchaser's computer system.

ANSI/UL 2900-1-2023

JULY 5, 2017
(Title Page Reprinted: April 14, 2023)

ANSI/CAN/UL 2900-1:2023

Standard for Software Cybersecurity for Network-Connectable Products,

Part 1: General Requirements

First Edition

July 5, 2017

This ANSI/UL Standard for Safety consists of the First Edition including revisions through April 14, 2023.

The most recent designation of ANSI/UL 2900-1 as an American National Standard (ANSI) occurred on April 14, 2023. ANSI approval for a standard does not include the Cover Page, Transmittal Pages, Title Page, Preface or SCC Foreword.

This standard has been designated as a National Standard of Canada (NSC) on April 14, 2023.

No Text on This Page

CONTENTS

APPENDIX D (INFORMATIVE)

Preface

This is the First Edition of the ANSI/CAN/UL 2900-1, Standard for Software Cybersecurity for Network-Connectable Products, Part 1: General Requirements.

ULSE Inc. is accredited by the American National Standards Institute (ANSI) and the Standards Council of Canada (SCC) as a Standards Development Organization (SDO).

This Standard has been developed in compliance with the requirements of ANSI and SCC for accreditation of a Standards Development Organization.

This ANSI/CAN/UL 2900-1 Standard is under continuous maintenance, whereby each revision is approved in compliance with the requirements of ANSI and SCC for accreditation of a Standards Development Organization. In the event that no revisions are issued for a period of four years from the date of publication, action to revise, reaffirm, or withdraw the standard shall be initiated.

In Canada, there are two official languages, English and French. All safety warnings must be in French and English. Attention is drawn to the possibility that some Canadian authorities may require additional markings and/or installation instructions to be in both official languages.

Only metric SI units of measurement are used in this Standard. If a value for measurement is followed by a value in other units in parentheses, the second value may be approximate. The first stated value is the requirement.

Appendices A, B, C, and D, identified as informative, are for guidance and informational purposes only.

Comments or proposals for revisions on any part of the Standard may be submitted to ULSE at any time. Proposals should be submitted via a Proposal Request in ULSE's Collaborative Standards Development System (CSDS) at http://csds.ul.com.

This Edition of the Standard has been formally approved by the Technical Committee (TC) on Software Cybersecurity for Network-Connectable Products: General Requirements, TC 2900-1.

This list represents the TC 2900-1 membership when the final text in this standard was balloted. Since that time, changes in the membership may have occurred.

TC 2900-1 Membership

Name	Representing	Interest Category	Region
Ahmadi, Mike	M Ahmadi	General	USA
Alvarez, Edison	Becton Dickinson	Producer	USA
Barker, Davis	Stanley Black & Decker	Producer	USA
Biggs, Douglas	UL Solutions	Testing & Stds Org	USA
Chevalier, Mathieu	Genetec	Producer	Quebec, Canada
Cosman, Eric	OIT Concepts LLC	Non voting member	USA
Datko, Joshua	Cryptotronix	General	USA
Dawson, Joe	EWA-Canada (an Intertek Co)	Testing & Stds Org	Newfoundland, Canada
Deskurakis, John	Carrier	Producer	USA
Dischert, Larry	Johnson Controls, Inc./Building Solutions North America	Commercial / Industrial User	USA
Dutta, Ashim	Eaton India Innovation Center	Producer	India
Fitzgerald, Brian	Food & Drug Administration	Government	USA
Fogleman, Greg	Department of Veterans Affairs	Government	USA
Garrett, Michael	Garrett Technologies Inc	General	USA
Garvy, Patrick	Honeywell	Producer	USA
Gatz, Stephen	Whirlpool Corp.	Producer	USA
Griffith, Steve	NEMA	Non-voting member	USA
Hicken, Arthur	PARASOFT	Testing & Stds Org	USA
Hornberger, Richard	Phoenix Contact Services Inc	Commercial / Industrial User	USA
Lee, Simon	U S Consumer Product Safety Commission	Non-voting member	USA
Leinonen, Juuso	ECRI	Supply Chain	USA
Li, Xiaodong	Lenovo (Beijing) Ltd	Producer	China
Martin, Robert	The Mitre Corporation	Commercial / Industrial User	USA
Maxey, Derek	Lockheed Martin Missiles & Fire Control	Supply Chain	USA
Mendoza, Ernesto	Signify North America Corporation	Producer	USA
Prince, Deborah R.	UL Standards & Engagement	TC Chair – Non-voting	USA
Rowland, Michael	IAEA	Government	Austria
Shkolnik, Moti	Firedome	Supply Chain	USA
Thayer, Rodney	Smithee Solutions LLC	General	USA
Tran, Phat	BC Safety Authority	Non-voting member	British Columbia, Canada
Treuthardt, Caroline	UL Standards & Engagement	TC Project Manager – Non-voting	USA
Vasserman, Eugene	Kansas State University	General	USA
Wang, Hui	CNCERT/CC	General	China
Wyman, Richard	CS 7 CONSULTING	General	USA

International Classification for Standards (ICS): 35.030, 35.110, 35.240.50

This Standard is intended to be used for conformity assessment.

The intended primary application of this standard is stated in its scope. It is important to note that it remains the responsibility of the user of the standard to judge its suitability for this particular application.

CETTE NORME NATIONALE DU CANADA EST DISPONIBLE EN VERSIONS FRANÇAISE ET ANGLAISE

No Text on This Page

INTRODUCTION

1 Scope

1.1 This standard applies to network-connectable products that shall be evaluated and tested for vulnerabilities, software weaknesses and malware.

1.2 This standard describes:

a) Requirements regarding the software developer (vendor or other supply chain member) risk management process for their product.

b) Methods by which a product shall be evaluated and tested for the presence of vulnerabilities, software weaknesses and malware.

c) Requirements regarding the presence of security risk controls in the architecture and design of a product.

1.3 This standard does not contain requirements regarding functional testing of a product. This means this standard contains no requirements to verify that the product functions as designed.

1.4 This standard does not contain requirements regarding the hardware contained in a product.

2 Normative References

2.1 All references are for the latest published version of the document, unless stated otherwise.

[1] Deleted

[2] Deleted

[3] Standard for Test Access Port and Boundary-Scan Architecture,
IEEE 1149

[4] Cybersecurity information exchange – Vulnerability/state exchange – Common vulnerabilities and exposures (CVE); retrievable from https://cve.mitre.org/,
ITU-T X.1520

[5] Cybersecurity information exchange – Vulnerability/state exchange – Common vulnerability scoring system (CVSS); retrievable from https://nvd.nist.gov/vuln-metrics/cvss,
ITU-T X.1521

[6] Cybersecurity information exchange – Vulnerability/state exchange – Common weakness enumeration (CWE),
ITU-T X.1524

[7] Cybersecurity information exchange – Vulnerability/state exchange – Common weakness scoring system (CWSS); retrievable from https://cwe.mitre.org/cwss,
ITU-T X.1525

[8] Cybersecurity information exchange – Event/incident/heuristics exchange – Common attack pattern enumeration and classification (CAPEC); retrievable from https://capec.mitre.org,
ITU-T X.1544

[9] Common Weakness Risk Analysis Framework (CWRAF); retrievable from https://cwe.mitre.org/cwraf/

[10] CWE/SANS Top 25 Most Dangerous Software Errors; retrievable from cwe.mitre.org/top25

[11] CWE On the Cusp: other weaknesses to consider; retrievable from *https://cwe.mitre.org/top25/cusp.html*

[12] OWASP Top 10; latest version retrievable from *https://www.owasp.org/index.php/Top_10_2013-Top_10*

[13] Information technology – Trusted platform module library,
ISO/IEC 11889

[14] Information technology – Security techniques – Digital signature scheme giving message recovery,
ISO/IEC 9796 (all parts)

[15] Information technology – Security techniques – Message Authentication Codes (MACs),
ISO/IEC 9797 (all parts)

[16] Information technology – Security techniques – Entity authentication,
ISO/IEC 9798 (all parts)

[17] Information technology – Security techniques – Hash-functions,
ISO/IEC 10118 (all parts)

[18] Information technology – Security techniques – Key management,
ISO/IEC 11770 (all parts)

[19] Information technology – Security techniques – Digital signatures with appendix,
ISO/IEC 14888 (all parts)

[20] Information technology – Security techniques – Cryptographic techniques based on elliptic curves,
ISO/IEC 15946 (all parts)

[21] Information technology – Security techniques – Encryption algorithms,
ISO/IEC 18033 (all parts)

[22] Information technology – Security techniques – Authenticated encryption,
ISO/IEC 19772 (all parts)

[23] The National Institute of Standards and Technology Cybersecurity Framework,
NIST

[24] Annex A: Approved Security Functions,
NIST SP 800-140C

[25] Annex D: Approved Key Establishment Techniques,
NIST SP 800-140D

[26] The National Institute of Standards and Technology Special Publication,
NIST 800-53

[27] Guidelines for Media Sanitization,
NIST SP 800-88

[28] Guide to Protecting the Confidentiality of Personally Identifiable Information,
NIST SP 800-122

[29] Federal Information Processing Standards – Digital Signature Standard (DSS),
FIPS 186-4

3 Glossary

3.1 ATTACK – The use of one or more exploit(s) by an adversary to achieve one or more negative technical impact(s).

3.2 ATTACK PATTERN – A description of a generic method for carrying out attacks.

3.3 AUTHENTICATION – The process of verifying the identity of an entity.

3.4 AUTHENTICITY – The property that data, information or software originate from a specific entity.

3.5 AUTHORIZATION – The process of giving an entity permission to access or manipulate the product, or the property that an entity has such permission.

3.6 BINARY CODE – Machine instructions and/or data in a format intended for a specific processor architecture.

3.7 BYTECODE – Instructions and/or data that are created from source code as an intermediate step before generating binary code. Bytecode is independent of a specific processor architecture and is typically handled by a virtual machine or interpreter.

3.8 COMMON ATTACK PATTERN ENUMERATION AND CLASSIFICATION (CAPEC) – Specified in ITU-T X.1544 (ref. [8]), the CAPEC is a publicly available resource providing a list and classification of a large number of attack mechanisms based on the topology of the environment.

3.9 COMMON VULNERABILITIES AND EXPOSURES (CVE) – Specified in ITU-T X.1520 (ref. [4]), the CVE is a publicly available resource providing common identifiers for known vulnerabilities and exposures.

3.10 COMMON VULNERABILITY SCORING SYSTEM (CVSS) – Specified in ITU-T X.1521 (ref. [5]), the CVSS is a publicly available resource providing a means for prioritizing vulnerabilities in terms of exploit potential.

3.11 COMMON WEAKNESS ENUMERATION (CWE) – Specified in ITU-T X.1524 (ref. [6]), the CWE is a publicly available resource providing a structured means to exchange unified, measurable sets of information providing common identifiers for software weaknesses, as well as consequences, detection methods and examples of each weakness.

3.12 COMMON WEAKNESS SCORING SYSTEM (CWSS) – Specified in ITU-T X.1525 (ref [7]), the CWSS is a publicly available resource providing a means for prioritizing CWEs based on their technical impact, ease of attack, and other factors.

3.13 COMMUNICATION PROTOCOL – A system of rules regarding syntax, semantics, synchronization and error recovery of data communication, allowing two or more entities to exchange information.

3.14 CONFIDENTIALITY – The property that data, information or software is not made available or disclosed to unauthorized individuals, entities, or processes.

3.14A CREDENTIALS – A set of attributes that uniquely identifies a system entity such as a person, an organization, a service, a role, or a device.

3.15 EXECUTABLE – A file containing instructions in binary code, which can be used by a computer to perform computational tasks.

3.16 EXPLOIT – An input or action designed to take advantage of a weakness (or multiple weaknesses) and achieve a negative technical impact.

NOTE: The existence of an exploit targeting a weakness is what makes that weakness a vulnerability.

3.17 EXTERNAL INTERFACE – An interface of the product that is designed to potentially allow access to an entity outside the product; for example user interfaces, remote interfaces, local interfaces, wireless interfaces and file inputs.

3.18 GENERATIONAL MALFORMED INPUT TESTING – A method of deriving malformed input test cases by using detailed knowledge of the syntax and semantics of the specifications of the protocol or file format being tested.

3.19 HARM – Physical injury or damage to the health of people, or damage to property or the environment.

3.20 I2C BUS – An inter-integrated circuit bus.

3.21 INTEGRITY –t he assurance that data can only be altered by authorized entities.

3.22 JTAG – Joint Test Action Group (JTAG) method of connection described in IEEE 1149, Standard for Test Access Port and Boundary-Scan Architecture.

3.23 KNOWN VULNERABILITY – A vulnerability described in the National Vulnerability Database (NVD).

NOTE: The NVD is accessible at https://nvd.nist.gov.

3.24 LIBRARY – A software set of code, functions, classes, procedures, scripts, configuration data that can be used by an executable

3.25 MALFORMED INPUT TESTING – A black-box testing technique used to reveal software weaknesses in a product by triggering them with invalid or unexpected inputs on the external interfaces of the product.

3.26 MALFORMED INPUT TEST CASE – The basic unit of malformed input testing, which consists of a single interaction with the product under test.

3.27 MALWARE – Software designed with malicious intent to disrupt normal function, gather sensitive information, and/or access other connected systems.

3.28 NETWORK CONNECTABLE – Any device, component, or software that can be connected via physical, wireless, cellular, and other non-physical transmission means to another device, component or software or groups of devices, components or systems of software.

3.29 PENETRATION TESTING – A mechanism of evaluation of a product to exploit vulnerabilities and weaknesses discovered in the vulnerability assessment phase.

3.30 PERSONALLY IDENTIFIABLE INFORMATION (PII) – Any information about an individual maintained by the product, including any information that can be used to distinguish or trace an individual‘s identity, such as name, social security number, date and place of birth, or biometric records;

AND

Any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information.

NOTE: This can be, but is not limited to an individual's location, health records and/or financial records that when used can determine the actual individual's identity.

3.31 PRODUCT – The network-connectable device, software or system under test.

3.32 PROTOCOL – See COMMUNICATION PROTOCOL

3.33 REMOTE INTERFACE – An external interface potentially allowing access to individuals, entities or processes regardless of geographic distance to the product.

3.34 REMOTE ACCESS – Access to the product via a remote interface.

3.35 RISK – The potential for harm or damage, measured as the combination of the likelihood of occurrence of that harm or damage and the impact of that harm or damage.

3.36 RISK ANALYSIS – The systematic use of available information to identify threats and to estimate risk.

3.37 RISK CONTROL – Any action taken or feature implemented to reduce risk.

3.38 RISK MANAGEMENT – Systematic application of management policies, procedures and practices to the tasks of analyzing, evaluating, controlling and monitoring risk.

3.39 SECURE ELEMENT – A tamper-resistant platform like a chip capable of securely hosting applications and their confidential and cryptographic data and that will prevent unauthorized access.

3.40 SECURITY – The process of having acceptable levels of confidentiality, integrity, authenticity and/or availability of product data and/or functionality through risk analysis.

3.41 SENSITIVE DATA – Sensitive data is any critical security parameter that can compromise the use and security of the product such as passwords, keys, seeds for random number generators, authentication data., personally identifiable information and any data whose disclosure could jeopardize the security properties of the product.

3.42 SOFTWARE – All pre-loaded data which creates, affects, and/or modifies the functionality of the product. This includes, but is not limited to, firmware, scripts, initialization files, pre-compiled code and interpreted code. This does not include software preloaded and programmed in an IC chip for small functions that require physical access and removal of the IC chip for reprogramming.

3.42A SOFTWARE BILL OF MATERIALS (SBOM) – A nested inventory, a list of ingredients that make up software components.

3.43 SOFTWARE WEAKNESS – A possible flaw in the architecture, design, coding, build process or configuration of software in the product that may render the product vulnerable to a security exploit.

3.44 SOURCE CODE – Computer instructions written in a human-readable high-level computer language, usually as text, including possible comments.

3.45 SPI – is a serial peripheral shared interface bus.

3.46 STATIC ANALYSIS – A process in which source code, bytecode or binary code is analyzed without executing the code.

3.47 TEMPLATE MALFORMED INPUT TESTING – Generates test cases by introducing anomalies into a valid message or file. Template malformed input test cases are not protocol aware and therefore will not contain items such as correct checksums and valid session IDs.

3.48 THREAT – A potentially successful attack, utilizing specific techniques and resources to take advantage of specific vulnerabilities or lack of risk controls within a product.

3.49 TRUSTED PLATFORM MODULE – Defines the requirements for a dedicated microprocessor with requirements for storage of cryptographic keys used to secure physical products and the software contained.

3.50 USER – A person or process using a product or accessing it over one of its external interfaces.

3.51 VENDOR – The manufacturer, reseller or supplier of a product, that takes final responsibility for the cybersecurity of that product towards the purchaser and/or user and which submits that product for testing according to this standard.

3.52 VULNERABILITY – A weakness identified in the product for which an exploit does exist, such that it can be directly used by an attacker.

3.53 WIRELESS INTERFACE – An external interface using electromagnetic waves, rather than some form of wire, to carry communication signals to and from a product.

4 Documentation Of Product, Product Design And Product Use

4.1 *Title deleted*

4.1 The vendor shall provide the following for a product evaluation:

a) A description of all functions (such as operational features, security, management functions, and the like), provided by the product;

b) A list of all external interfaces of the product in its intended configuration, along with all communication protocols supported on each of these interfaces, where applicable, including:

1) All remote interfaces;

2) All local interfaces – product local interfaces such as SPI, I2C, JTAG and serial ports;

3) All wireless interfaces;

4) All external file inputs;

NOTE: Supported means capable of being enabled.

c) A list of all executables and libraries in the product, including all third party and open-source software. All executables and libraries shall be identified by both a software name and version number. Known operating system executables and libraries can be defined as the stated distribution of that operating system but any additional operating system libraries not defined in the known distribution shall be identified.

NOTE: An equivalent software bill of materials (SBOM) i.e a list of the contents of the software can be substituted.

d) The existing source code of all software in the product that is available including the scripts, libraries, makefiles, and build configuration parameters necessary to replicate the production build environment shall be provided as needed per the section Software Weakness.

NOTE: The specified materials are intended to allow an evaluator to establish and configure the product so it can be deployed and function correctly in its intended use environment. This requirement does not preclude developers from also providing representative pre-built, pre-configured products for testing purposes.

e) The binary code and/or bytecode and associated identifiers of all software in the product, unless the vendor has no access or no rights to this binary or bytecode as in a third-party library that is controlled. A risk management assessment of a controlled third-party library that the vendor has no access or rights to shall be required. The binary code and/or bytecode provided shall be unobfuscated when available.

NOTE: An associated identifier such as a hash or signature or SWID tags that can validate the contents and functionality of the binary and/or bytecode.

f) Detailed instructions on the product software build and integration process.

g) A clear definition of the boundary between the product and elements of the system that are outside the scope of evaluation. See 6.8.

NOTE: If the objective is to do a sub-component or a system within a system, the boundary between the products in and out of scope need to be well defined.

h) Any run-time configuration files required for the operation of the product and constituent software under evaluation.

NOTE: When applying this standard, it is intended that a product boundary be identified that includes all product components utilized to meet applicable UL 2900-1 requirements.

5 Product Design Documentation

5.1 The vendor shall provide the following for a product evaluation:

a) The security risk analysis for the product as described in 12.1 of this standard.

b) The design documentation containing sufficient details to allow an evaluation of the way each of the risk controls mentioned in Sections 7 – 11 is implemented in the product.

6 Documentation for Product Use

6.1 Product use documentation supports the overall cyber security objectives through the product life cycle.

NOTE: Other organizations have written various good practices manuals. The National Institute of Standards and Technology Cybersecurity Framework, and the National Institute of Standards and Technology Special Publication 800-53 series are two examples.

6.2 The vendor shall provide documentation addressing security considerations on the intended use of the product and its configuration.

6.3 The vendor shall provide documentation addressing the environment in which the product is intended to be used.

6.4 The vendor shall provide instructions to ensure the effectiveness of security functions and controls during product use.

6.5 The vendor shall document all external interfaces and all communication protocols used externally by the product, including which external interfaces support which protocols.

6.6 The vendor shall provide documentation of all version numbers of all software binaries, libraries and executables used in the product. Known operating system executables and libraries can be defined as the stated distribution of that operating system but any additional operating system libraries not defined in the known distribution shall be identified.

6.7 The vendor shall provide documentation listing security-related event descriptions, logged by the product according to 11.3 and 11.4.

6.8 The vendor shall provide documentation of requirements and recommendations on the product's configuration and the environment in which the product is installed that are necessary to ensure the product's security.

NOTE: This should include requirements on network security, physical access control to the product, firewall ports and protocols, local interfaces' configuration options etc.

6.9 The vendor shall provide documentation that the product's authentication and authorization methods and subsequent authenticated and authorized communications cannot be bypassed using any procedure that uses less computation than exercising all elements of the set of values necessary for systematic deduction of the authentication's secret value(s).

6.10 Any overrides to 6.9 shall be evaluated and documented in the risk assessment with a rationale.

NOTE 1: For example, if a key is used for authentication, then it should require at least as many operations to circumvent the authentication means as it does to guess the credential.

NOTE 2: This requirement is intended to define disclosure requirements stating the difficulty of bypassing, brute-forcing, or otherwise circumventing the authentication system of the product.

NOTE 3: If authenticated communications is not physically or cryptographically secure, then 6.9 cannot be met.

7 Risk Controls

7.1 General

7.1.1 The product (or the product's vendor, as applicable) shall comply with all of the security risk controls specified in Sections 8 – 11, unless the risk assessment performed by the vendor according to Section 12, Vendor Product Risk Management Process, shows that the risks associated with not implementing a specific control are acceptable in product use.

7.1.2 If the vendor chooses to not comply with one or more of these risk controls, the vendor shall document and justify this in the risk analysis per 12.1.

7.1.3 Any time sensitive functionality shall be evaluated and documented in the risk assessment with a rationale.

7.1.4 Functionality that relies on time keeping and/or time synchronization shall be evaluated in the risk assessment with a rationale.

7.1.5 Product use cases or updates to the product that have the potential to change the security risk, shall be evaluated and included in the risk analysis.

8 Access Control, User Authentication and User Authorization

8.1 Product operation or management functions which may affect or alter the security of the product as defined by the vendor documentation shall require authentication and authorization prior to access of the product need to be documented as per Section 12, Vendor Product Risk Management Process.

8.2 Authentication services to the product shall implement an inactivity time-out or other appropriate mechanism to prevent perpetual authorization. The inactivity time-out interval shall be configurable at a user level or by the product's response to an event or action.

8.3 If the product uses an authentication credential mechanism for authenticating users:

a) The product shall use a cryptographically secure mechanism complying with the requirements in Appendix B to store and transmit the credential.

b) Authentication error messages provided by the product shall not allow for enumerating valid credentials.

c) The product shall support the possibility to set requirements regarding the, complexity, update frequency, strength or length for credentials with the following rules:

i) If the credential uses a password, its minimum length shall be 6 characters.

ii) For every 10 sequential unsuccessful authentication attempts of a user, operator or process within the product over a one-hour period, the credential shall either be disabled or a timeout of a minimum of 30 minutes shall be applied before another authentication attempt is allowed.

iii) If i) or ii) are not met, the required minimum length, frequency and strength of the credential shall be evaluated and documented in the risk assessment. Some alternatives that can be utilized are an increasing delay for each unsuccessful attempt and anti-robot protection such as captcha tests.

NOTE: A complexity test can also be run. Complexity options can include special characters, minimum length, upper and lowercase and combinations of options and/or key sizes.

d) The product shall protect against brute force attacks.

NOTE: Examples of mechanisms to do so include key stretching; salts or preventing login attempts for the given credential after a specified number of failed attempts and/or dictionary attacks and/or rainbow table use.

e) The product shall have no default credential that cannot be modified or supplanted by an alternative (like a user defined credential that replaces a built-in factory default). All default credentials should have a mechanism for change upon first use after installation with a user notification of Default Credentials in use if applicable.

f) The product shall have an option to limit the number of unsuccessful attempts.

8.4 For products using a role-based access mechanism:

a) The vendor shall clearly document all existing roles and their associated privileges.

b) There shall be an 'administrator' or 'system' role that has privileges exclusively related to the management of the product. Such privileges shall not be granted to other roles.

NOTE: Other roles are to be assigned using the principle of least privilege.

8.5 The product shall support the possibility to manage the list of valid user accounts by adding, removing and/or suspending user accounts (e.g. “allow-listing” and “block-listing”) or by adding, revoking, or updating of authentication credentials.

8.6 The product shall support assigning privileges and permissions to roles and credentials.

8.7 The product shall enforce the principle of least privilege for every authorized role or user that can be authenticated.

8.8 When an authenticated session is terminated, the product shall ensure the renewed session is authenticated prior to allowing access. Stored data from the previous session shall not be used to simplify/ease/partially bypass authentication mechanisms during a new session creation.

NOTE: The session is the protected communication once authentication has occurred. Protected means compliant with the Standard.

8.9 If the product uses other mechanisms for authentication besides username and password, the mechanism shall be such that the product’s authentication method cannot be bypassed using any procedure that uses less computation than exercising all elements of the set of values necessary for systematic deduction of the authentication’s secret value(s).

NOTE 1: For example, if a key is used for authentication, then it shall require at least as many operations to circumvent the key as it is to determine the key.

NOTE 2: This requirement is intended to address the difficulty of bypassing, brute-forcing, or otherwise circumventing the authentication system of the product.

9 Remote Communication

9.1 The product shall ensure the integrity and authenticity of all data communicated over any remote interface. For this, the product shall use security functions complying with the requirements in Appendix C.

Exception: Remote interfaces that report status, do not provide command and control functionality or do not transmit sensitive data, etc., may not require integrity and authenticity but will need to be documented as per Section 12, Vendor Product Risk Management Process.

NOTE: Remote interface describes interaction points outside the defined boundaries of the product subject to the requirements of this standard.

10 Sensitive Data

10.1 The product shall ensure the confidentiality of all sensitive and personally identifiable data information generated, stored, used, or communicated by the product. The product shall use a secure mechanism complying with the requirements in Appendix B to store the sensitive data and personally identifiable information.

10.2 For the purposes of 10.1, the vendor shall identify and document which data is to be considered sensitive.

10.3 The product shall utilize only cryptographic algorithms listed in Appendix C for any security protocol.

10.4 The product shall use a separate cryptographic key for each service, operation, or function (e.g. data at rest encryption, transport layer encryption, operator role authentication, remote software upgrade image integrity). The vendor shall clearly document the intended purpose of each key used by the product. Rationale shall be documented in accordance with Vendor Product Risk Management Process, Section 12.

NOTE: Purposes may include (but are not limited to) data encryption, providing data authenticity and integrity, key wrapping, random number generation or digital signatures.

11 Product Management

11.1 The product shall be designed and implemented to allow for application of security updates to the product's software. This process will also support reverting to previously installed version if the update fails. The roll back would revert to the previously installed version.

11.2 The product shall verify the authenticity and integrity of any software update cryptographically, before installing the update. Product updates shall be possible in an offline environment. This offline product update mode should also still support validation of authenticity and integrity.

11.3 The product shall be capable of maintaining one or more log(s) of all security-related events, such as successful and unsuccessful login attempts, change of user authentication credentials, changes in the list of valid user accounts, successful and unsuccessful software updates, etc.

11.4 Unless and until they are transmitted to an external data storage, the product shall store all security-related logs in non-volatile memory and shall not allow non-privileged users to remove or change them.

11.5 Decommissioning of the product after its use shall allow the ability to completely erase all user defined:

a) Configuration data;

b) Sensitive data; and

c) Personally identifiable information.

NOTE: NIST SP 800-88 may be used as a reference. As an example, zeroing the data is acceptable. Removal of data from the product can be performed as an operation or as a process procedure.

11.6 The following are approved integrity mechanisms for software updates. Validating software updates OR software source using the following techniques:

a) A message authentication code generated on the software and firmware binaries, executables and libraries.

b) A digital signature applied to the software and firmware components.

c) A hash applied to the software and firmware binaries, executables, and libraries, where the hash is published in such a way that it is possible for the product to securely identify the source of the software and its contents.

11.7 All integrity mechanisms defined in 11.6 shall comply with Appendix C.

11.8 Documentation shall exist describing the plan for providing validated software updates and patches as needed throughout the lifecycle of the product to continue to assure its continued security management.

RISK MANAGEMENT

12 Vendor Product Risk Management Process

12.1 When designing the product, the vendor shall establish and document a security risk analysis for the product, containing:

a) An identification of all product functionalities and all sensitive data stored, processed or used by the product;

b) A list of identified threats for the product, its functionalities and data;

c) An assessment of the impact of each identified threat, its operational impact and any or sensitive data that would be exposed, should it become a reality;

d) An assessment of the likelihood of each identified threat based on the vendor's assumptions;

e) A determination of the resulting vendor defined risk level for each threat, considering its impact and likelihood;

f) Risk management criteria, (i.e. clear criteria to determine whether or not a given vendor defined risk level is acceptable);

g) A determination of suitable risk controls to mitigate each identified threat with an unacceptable risk level. All risk controls in Sections 7 – 11 shall be considered. The vendor shall identify any additional risk controls that need to be implemented to mitigate identified threats. The vendor shall create a traceability matrix showing the relationship between identified threats and implemented risk controls.

h) An assessment of the residual risk level for each identified threat after application of these risk controls.

12.2 When carrying out the threat analysis resulting in the list meant in 12.1(b), the vendor shall make use of a classification scheme for risks identified. Some common examples that may be used are:

a) The Common Attack Pattern Enumeration and Classification, (CAPEC; see ref. [8]) in order to convey the completeness of the analysis;

b) DREAD model of Damage, Reproducibility, Exploitability, Affected users, Discoverability;

c) Equivalent Risk classification methodology.

12.3 The vendor shall document a risk evaluation method for the possible presence of known (types of) vulnerabilities in the product. This method shall describe the criteria that the vendor will use to evaluate the level of risk for each (type of) known vulnerabilities that may be found in product. The method shall also establish the level below which a risk is acceptable to the vendor. The evaluation criteria shall be based on risk factors including, but not limited to, the CVSS score of the vulnerability, the intended use of the product and the environment in which the product would be used. If the vendor has allowed for the presence of any known vulnerabilities in the product, the vendor's security risk analysis for the product shall contain a description of each accepted known vulnerability:

a) CVE standard vulnerability identifier;

b) The software location of the vulnerability;

c) A risk analysis, performed and documented according to the method and criteria, documented by the vendor as their risk evaluation method for known vulnerabilities in the product, showing that the risk level associated to the presence of this vulnerability is acceptable.

12.4 *Combined with 12.3*

12.5 The vendor shall likewise document a risk evaluation method for the possible presence of known (types of) software weaknesses in the product. This method shall describe the criteria that the vendor will use to evaluate the level of risk for each (type of) software weakness that may be found in product. The vendor shall make use of the Common Weakness Risk Analysis Framework (CWRAF), ref. [9], or comparable, as part of a strategy of risk management. The method shall also establish the level below which a risk is acceptable to the vendor. The evaluation criteria shall be based on risk factors including, but not limited to, the CWSS score of the weakness, the intended use of the product and the environment in which the product would be used. In case the vendor is aware of and has accepted the presence of any software weaknesses in the product, the vendor's security risk analysis for the product shall contain a description of each accepted weakness:

a) CWE standard weakness identifier;

b) The software location of the weakness;

c) A risk analysis, performed and documented according to the vendor's documented risk evaluation method for known vulnerabilities in the product, showing that the risk level associated with the presence of this weakness is acceptable;

d) External compensating controls to help further reduce the residual risk.

12.6 *Combined with 12.5*

12.7 To verify compliance with 12.1 – 12.3 and 12.5, the security risk analysis for the product shall be evaluated along with the product design documentation. In particular, the following shall be determined:

a) Sufficient coverage during the security risk analysis with regard to the identification of product functionality and data and threats, impact, likelihood and resulting risk.

b) Sufficient adoption of risk controls listed in Sections 7 – 11 by either implementing each control in the product or justifying why the risk level of not implementing a control is acceptable.

c) Sufficient implementation of risk controls per the requirements in Sections 7 – 11.

NOTE: Sufficiency is established via analysis of traceability through the risk management process.

12.8 To verify compliance with 12.3 and 12.5, the vendor's risk evaluation methods for the presence of known vulnerabilities and software weaknesses shall be evaluated.

SOFTWARE COMPOSITION

13 Known Vulnerability Testing

13.1 *Deleted*

13.2 *Deleted*

13A Software Composition Analysis

13A.1 The vendor shall decompose the software system to the level required to perform an analysis of weaknesses and vulnerabilities in accordance with Sections 17 and 19 below.

NOTE: This process may also be known as Software Composition Analysis. Information produced from this can be used to build a software bill of materials (SBOM).

14 Malware Testing

14.1 The binary code and bytecode in the product shall be scanned by at least one malware detection tool to identify if any known malware exists in the final deliverables of the product. The malware tools shall be applicable to the operating system that the software resides on.

14.2 To verify compliance with 14.1, all available binary code and bytecode provided by the vendor shall be inspected for known malware.

Exception: If a malware detection tool is not available for the applicable operating system, a risk assessment shall be conducted and documented per Section 12.

15 Malformed Input Testing

15.1 The product shall continue to operate as intended when subject to invalid or unexpected inputs on its external interfaces and shall not display unexpected behavior, such as, but not limited to the following:

a) The product resets or reinitializes its configuration;

b) *Deleted*

c) A process hangs;

d) The testing uses resources of the product and the product does not relinquish these resources after testing;

e) The product software throws an unhandled exception;

f) A storage data corruption occurs;

g) The product loses the connection to the malformed input testing tool;

h) The specified behavior of the product is interrupted and the product does not continue to operate as intended within 2 minutes or a timeframe defined by the manufacturer;

i) The product shall not disclose any personally identifiable information or sensitive data over any interface enumerated in 4.1(b).

j) The product shall not become non-responsive on external interfaces other than the one under test by the input testing tool.

15.2 To verify compliance with 15.1, malformed input testing shall be performed on the product as described in this section.

15.3 During malformed input testing, the product shall be configured per vendor documentation provided per 6.1 and 6.6.

15.4 Malformed input testing shall take place within a representative environment per vendor documentation provided per 6.1 and 6.6.

15.5 The product shall be inspected to verify the presence of those and only those external interfaces specified in the vendor's documentation per 4.1(b).

15.6 The product shall be subjected to malformed input testing of all file inputs, all remote interfaces, using all protocols supported by the product on these interfaces as listed by the vendor per 4.1(b). Each protocol on an interface shall be subjected to generational malformed input testing when available. Template malformed input testing may be used if a protocol is proprietary and there are no generational malformed input tools available for that protocol.

15.7 If generational malformed input testing is used for a protocol, testing shall evenly apply to all fields of the protocol and the testing shall implement exception handling based on the context of the protocol.

NOTE: Exception handling includes checks on message lengths, message identifiers, integrity checks, correct use of cryptographic protocols and other critical protocol attributes.

15.8 If template malformed input testing is used for a protocol on an interface, the template used shall cover all fields of the protocol.

15.9 If generational malformed input testing is applied for a protocol on an interface, at least 1,000,000 unique and independent test cases or a minimum of 8 hours of test case execution shall be carried out, whichever comes first.

15.10 If template malformed input testing is applied on an interface, at least 5,000,000 unique and independent test cases or a minimum of 8 hours of test case execution shall be carried out, whichever comes first.

15.11 If a protocol semantic structure space is substantially smaller than the limits set in 15.9 and 15.10 because of the lack of uniqueness or time to completion, the vendor shall supply the maximum range of unique cases and timeframe for execution to be used.

16 Structured Penetration Testing

16.1 The product under test shall have no known vulnerabilities known to the vendor that can be exploited and/or cause the product to crash, degrade, or perform in the manner not consistent with designed functionality without a recovery to its previous state after the test is completed in 2 minutes or within a timeframe defined by the manufacturer. The initial configuration shall be set up per the following:

a) To verify compliance with 16.1, penetration testing shall be carried out on the product as described in this section.

b) During penetration testing the product shall be configured per vendor documentation provided per 6.1 and 6.6.

c) Penetration testing shall take place within a representative environment per documentation provided per 6.1, 6.6, and 6.8.

16.2 The product shall be subjected to the penetration testing in order to try to find and exploit any flaws in the product based on the following conditions:

a) Circumvent the risk controls and security configuration of the product;

b) Attempt to engage the product in a denial of service;

c) Attempt to access and authenticate on the product via unauthorized means;

d) Attempt to exploit vulnerabilities acceptable in the risk analysis;

e) Attempt to elevate privilege on the product.

16.3 Attempts shall be made to identify system, application and service information via scanning of the ports, interfaces and services. Use of that information shall be considered in attempts to circumvent the security measures of the product. Exploit tools and scripts shall be used for discovered information to attempt to access the product, elevate the privilege once accessed or to gain further information about the product.

16.4 All risk assessment items scored in the risk assessment as not addressed shall be assessed with attempts to exploit to validate the risk assessment per Section 16, Structured Penetration Testing.

SOFTWARE WEAKNESSES

17 Software Weakness Analysis

17.1 The product or system under test shall contain no known software weaknesses that are unacceptable per Section 12.

NOTE 1: Typically, identified weaknesses must be eliminated from the product. However, in some cases, weaknesses can be shown to have a low enough risk that they can be acceptable. Paragraph 12.5 identifies the elements and actions required by the produced developer to determine if a weakness is acceptable. Any weaknesses that are identified as unaddressed will be tested through structured penetration testing (16.4) to verify they cannot be exploited.

NOTE 2: See Appendix A2 for information on weaknesses and vulnerabilities.

17.2 To verify compliance with 17.1, the product shall be evaluated using static source code analysis, and static binary and bytecode analysis as described in Sections 17 and 19. The testing should be conducted using the latest version of the sources mentioned in Appendix A, as a guide and as applicable to the product under testing.

NOTE: Organizations interested in a list of known high risk software weaknesses may consider ISO/IEC 5055:2021, MISRA C++:2008, or MISRA C:2012.

17.3 Static source code analysis shall be performed as follows:

a) All source code provided per 4.1(d) shall be evaluated by means of static code analysis;

b) The product shall be evaluated for at least all software weaknesses listed in the latest versions of the sources mentioned in Appendix A, as applicable to the product;

c) Selection of the tool should be determined by the strategy determined in 12.5.

18 Static Source Code Analysis

18.1 *Deleted*

18.2 *Deleted*