EDSA-310
ISA Security Compliance Institute –
Embedded Device Security Assurance –

Common requirements for communication robustness testing of IP-based protocol implementations

Version 1.7

September 2010

A. DISCLAIMER

ASCI and all related entities, including the International Society of Automation (collectively, “ASCI”)provide all materials, work products and, information (‘SPECIFICATION’) AS IS, WITHOUT WARRANTY AND WITH ALL FAULTS, and hereby disclaim all warranties and conditions, whether express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of reliability or availability, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence, all with regard to the SPECIFICATION, and the provision of or failure to provide support or other services, information, software, and related content through the SPECIFICATION or otherwise arising out of the use of the SPECIFICATION. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION, OR NON-INFRINGEMENT WITH REGARD TO THE SPECIFICATION.

WITHOUT LIMITING THE FOREGOING, ASCI DISCLAIMS ALL LIABILITY FOR HARM TO PERSONS OR PROPERTY, AND USERS OF THIS SPECIFICATION ASSUME ALL RISKS OF SUCH HARM.

IN ISSUING AND MAKING THE SPECIFICATION AVAILABLE, ASCI IS NOT UNDERTAKING TO RENDER PROFESSIONAL OR OTHER SERVICES FOR OR ON BEHALF OF ANY PERSON OR ENTITY, NOR IS ASCI UNDERTAKING TO PERFORM ANY DUTY OWED BY ANY PERSON OR ENTITY TO SOMEONE ELSE. ANYONE USING THIS SPECIFICATION SHOULD RELY ON HIS OR HER OWN INDEPENDENT JUDGMENT OR, AS APPROPRIATE, SEEK THE ADVICE OF A COMPETENT PROFESSIONAL IN DETERMINING THE EXERCISE OF REASONABLE CARE IN ANY GIVEN CIRCUMSTANCES.

B. EXCLUSION OF INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL ASCI OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL,PUNITIVE, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF PROFITS OR CONFIDENTIAL OR OTHER INFORMATION, FOR BUSINESS INTERRUPTION, FOR PERSONAL INJURY, FOR LOSS OF PRIVACY, FOR FAILURE TO MEET ANY DUTY INCLUDING OF GOOD FAITH OR OF REASONABLE CARE, FOR NEGLIGENCE, AND FOR ANY OTHER PECUNIARY OR OTHER LOSS WHATSOEVER) ARISING OUT OF OR IN ANY WAY RELATED TO THE USE OF OR INABILITY TO USE THE SPECIFICATION, THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT OR OTHER SERVICES, INFORMATON, SOFTWARE, AND RELATED CONTENT THROUGH THE SPECIFICATION OR OTHERWISE ARISING OUT OF THE USE OF THE SPECIFICATION, OR OTHERWISE UNDER OR IN CONNECTION WITH ANY PROVISION OF THIS SPECIFICATION, EVEN IN THE EVENT OF THE FAULT, TORT (INCLUDING NEGLIGENCE), MISREPRESENTATION, STRICT LIABILITY, BREACH OF CONTRACT OF ASCI OR ANY SUPPLIER, AND EVEN IF ASCI OR ANY SUPPLIER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Revision history

version	date	changes
1.4	2010.06.15	initial version published to http://www.ISASecure.org
1.7	2010.09.17	add approach for distributed I/O modules; refine treatment of digital vs. analog control output monitoring; add discrete control outputs; add sequences of invalid PDUs to basic robustness testing; use maximum auto-negotiated rate in load stress testing; added concept of device vendor test harness for monitoring control loop; increase minimum rate for TD; remove protocol conformance test criteria since other industry efforts cover conformance; separate tests using high but supported rate and full auto-negotiated link rate; remove discovery phase since not required for performing uniform testing over all devices; remove mixing of valid and invalid messages in load testing since valid messages create more load on device

Contents

Foreword

NOTE This is one of a series of communication robustness testing (CRT) specifications for embedded devices. This specification is the overarching document in the series. Other documents in the series that discuss CRT for specific protocols refer to this document using the symbolic links noted in 3.3. CRT is one of three elements required for ISASecure certification of embedded devices. The other two elements are Software Development Security Assessment (SDSA) and Functional Security Assessment (FSA). The full current list of documents related to embedded device security assurance can be found on the ISASecure web site http://www.ISASecure.org.

1 Scope

This document is intended to provide requirements for testing the robustness of embedded device implementations of IP-based protocols, as a measure of the extent to which such implementations defend themselves against

- correctly formed messages and sequences of such messages;
- single erroneous messages; and
- inappropriate sequences of messages;

where failure of the device to continue to provide concurrent automation system control and reporting functions, demonstrates potential security vulnerabilities within the device. The term used here for this type of testing is communication robustness testing (CRT). This document is not intended to serve as a guide for testing the correctness of implementations or conformance to mandatory provisions of the controlling standard(s), which cannot be determined solely by observing a device's response to external stimuli.

The goals of the test approach are to identify the presence of common programming errors and known denial of service vulnerabilities of networking protocols, which impact the robustness of embedded devices that use these protocols. Tests are specified to a level such that these goals are covered, although specific test data is not defined. These tests will not necessarily identify intentionally malicious code, nor is that a feasible goal for any practical testing regimen. Development process assurances such as the SDSA are required to mitigate the potential for introduction of malicious code.

This document covers requirements on test tools and test process that are common in the sense that they apply to testing for all protocols. It also defines requirements that determine how the set of protocols to be tested is selected, and criteria for passing the overall CRT element of the ISASecure EDSA (Embedded Device Security Assurance) certification.

2 Normative references

[EDSA-300] *ISCI Embedded Device Security Assurance – ISASecure Certification Requirements,* as specified at http://www.ISASecure.org

[PORT] *IANA port numbers*, as specified at http://www.iana.org/assignments/port-numbers

NOTE 1 For each RFC*nnn*, the controlling version can be found at http://tools.ietf.org/html/rfc*nnn*.

RFC1122, *Requirements for internet hosts – communication layers*

NOTE 2 Only 4.1 is referenced.

3 Definitions and abbreviations

3.1 Definitions

3.1.1
accessible network interface
network interface declared by the device vendor as suitable for use during normal operation, that supports for operation or instrumentation any protocol subject to CRT, and such that connection can occur without physical reconfiguration

NOTE Some network interfaces on embedded devices are internal connections only, and/or have physical protection intended to help prevent an external network connection to the ambient network.

3.1.2
adequately maintain essential service
maintain essential services at a level deemed suitable for an embedded control device while under a given type of attack

NOTE See definition below for essential service and 8.4.

3.1.3
core protocol
a protocol in the set ICMP, IPv4, ARP, IEEE 802.3 (Ethernet II), UDP or TCP

NOTE These protocols form the underlying infrastructure for many other protocols used in embedded devices. Once IPv6 and wireless are covered by the ISASecure certification, additional protocols will be added to this list.

3.1.4
discrete output
output that can assume a pre-defined, finite number of values (usually represented as small unsigned integers)

3.1.5
embedded device
special purpose device running embedded software designed to directly monitor, control or actuate an industrial process

NOTE 1 Attributes of an embedded device are: no rotating media, limited number of exposed services, programmed through an external interface, embedded operating system or firmware equivalent, real-time scheduler, may have an attached control panel, may have a communications interface. Examples of an embedded device are: PLC, field sensor devices, SIS controller, DCS controller.

NOTE 2 An embedded device with a dual or redundant configuration is considered a single device to be tested from the point of view of this specification.

3.1.6
erroneous (message or PDU or option)
PDU that violates either syntactic rules on PDU structure or semantic rules on PDU content or both, or PDU option that violates either syntactic rules on PDU option structure or semantic rules on PDU option content or both

NOTE 1 Semantic and syntactic rule violations can interact, as when the value of one field determines the size of another field.

NOTE 2 The term erroneous includes syntactic malformation, semantically invalid values, and contextually invalid values and sequences.

NOTE 3 This is addressed further in 9.3.2.

3.1.7
instrumentation protocol
implementation of a protocol used for device vendor development or test purposes but not required by end users

3.1.8
essential service
specified subset of the services provided by a device that is agreed between the applicant for certification and the test laboratory, at the start of certification

NOTE See 6.2. Essential services are a subset of the following 6 services: the process control/safety loop, process view, command, process alarms, provide essential history data and peer-to-peer control communication. The first four of these are always considered essential services.

3.1.9
jitter
the difference between the time a signal event is detected and the expected time based on a reference signal

3.1.10
measurement jitter
possible error in jitter measurement

3.1.11
operational mode
device state that is manually selected to allow access to particular device functions, such as configuration, control operations, update

NOTE Not all embedded controllers use the concept of operational modes.

3.1.12
peer-to-peer control communication
communication with another embedded device which ultimately may cause a change to the process parameters

3.1.13
testing device
conceptual single network-connected device, possibly consisting of multiple physical network-connected devices, used to test the robustness of the device under test

3.1.14
test laboratory
organization that is carrying out communication robustness testing for the ISASecure EDSA certification process

3.2 Abbreviations

The following abbreviations are used in this document

ARP	address resolution protocol
CRT	communication robustness testing
DCS	distributed control system
DoS	denial of service
DUT	device under test
EDSA	embedded device security assurance
IANA	Internet assigned numbers authority
ICMP	Internet control message protocol
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input/Output
IP	Internet (network layer) protocol
IPv4	IP version 4 (uses 32-bit network layer addresses)
IPv6	IP version 6 (uses 128-bit network layer addresses)
ISCI	ISA Security Compliance Institute
Ki	International Electrotechnical Commission standard symbol for the number 1024
(*N*)PDU	(*N*-layer) protocol data unit, where *N* = D (data-link), N (network), T (transport), A (application), etc
PLC	programmable logic controller
SIS	safety instrumented system
SYN	synchronize sequence numbers, a flag used in TCP packets during connection establishment
TCP	transmission control protocol

TD	testing device
UDP	user datagram protocol

3.3 Symbolic links to this document used by related documents

The following symbolic links are used by related documents to reference specific elements of this document. The references are symbolic so that editing of this document that changes this numbering will not impact all of the referencing documents.

Table 1 – Inter-document symbolic links and their local dereferencing

Symbolic link	Referenced elements
[CRT.Essential_services]	8.4.3
[CRT.Rate_limiting]	Requirement CRT.R18 and 9.3.5
[CRT.Reproducibility]	9.5
[CRT.Terminology_of_Erroneous]	9.3.2
[CRT.Test_configurations]	9.2

4 Types of tests and test order

This clause defines the types of tests that make up CRT and requirements on the sequence in which the tests are executed. Figure 1 illustrates these requirements.

Requirement CRT.R1 – Types of CRT tests

CRT for an embedded device SHALL consist of:

a) interface surface tests per the requirements of Clause 8; and

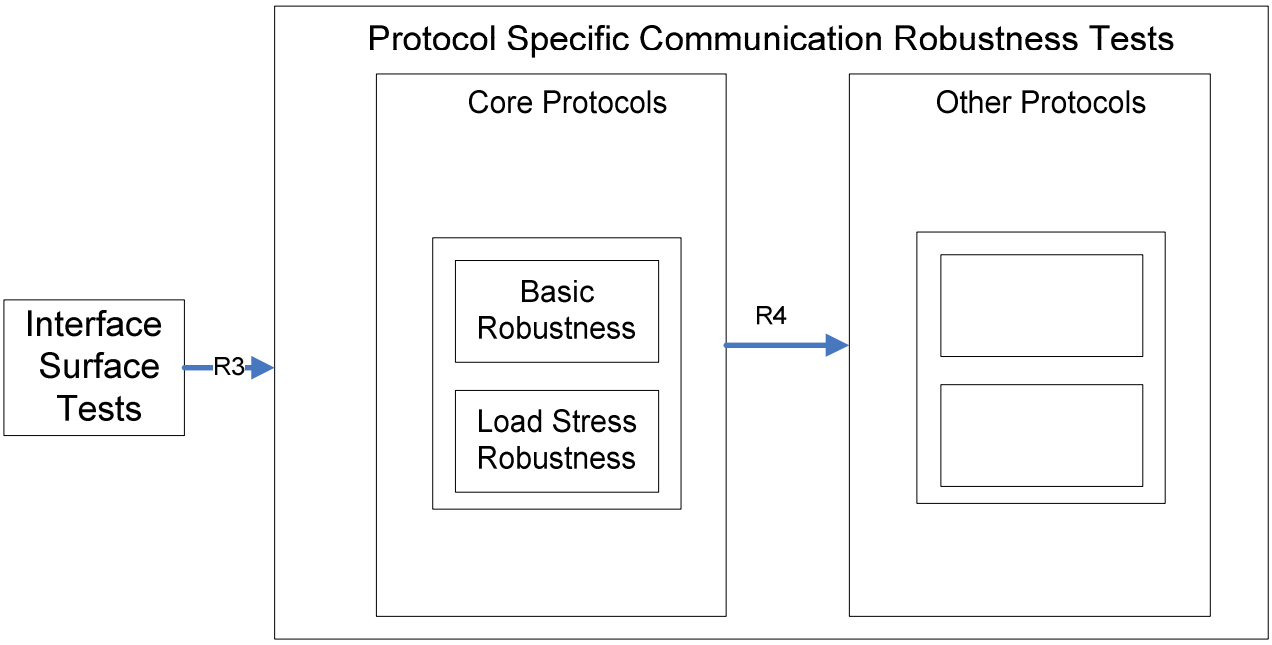
b) protocol specific robustness testing per the requirements of Clause 9 for all protocols listed in [EDSA-300] that are applicable to the device, where applicability is defined in Requirement CRT.R2.

Requirement CRT.R2 – Applicable protocols for CRT

The tests and assessments in a protocol-specific specification listed in [EDSA-300] SHALL be considered *applicable* to an embedded device if either:

- the interface surface test results or a DUT-based utility as described in Clause 8 of this specification conclude that a port associated with the subject protocol of that specification per the mapping in [PORT] may be active for operation or instrumentation on the device; or
- the vendor has stated that the protocol may be active for operation or instrumentation on the device per Requirement CRT.R17.

NOTE 1 CRT requirements and testing are available for a defined set of standard protocols under the ISASecure certification program. This set of protocols will evolve over time, and is maintained in [EDSA-300].



NOTE 2 The arrows in the figure correspond to requirements on test order.

Figure 1 – Order of execution for communication robustness test types

Requirement CRT.R3 – Interface surface tests precedence

Interface surface test cases as defined in Clause 8 SHALL be the first CRT tests performed.

Requirement CRT.R4 – Core protocol tests precedence

Protocol-specific robustness tests (Clause 9) for the protocols ICMP, IPv4, ARP, IEEE 802.3 “Ethernet”, (either as Ethernet II or as IEEE 802.3 Type 1 and IEEE 802 SNAP), UDP and TCP SHALL be performed before these tests for other protocols.

NOTE 3 The protocols listed form the underlying infrastructure for many other protocols used in embedded devices. Once IPv6 and wireless are covered by the ISASecure CRT, additional protocols will be added to this requirement.

5 CRT pass criteria

The requirements in this clause specify how a test laboratory will define for a given device, the criteria for passing the CRT.

Requirement CRT.R5 – Criterion for CRT pass

A test laboratory SHALL determine that the CRT for an embedded device has passed if the device submitted for certification passes all CRT tests listed in Requirement CRT.R1. If an embedded device as submitted for certification includes a distributed I/O module that itself has an accessible network interface, then both the overall device and the distributed I/O module SHALL be required to pass applicable tests.

Requirement CRT.R6 – Single configuration DUT

All tests and assessments required for CRT SHALL pass on one or more physical DUTs of identical configuration in order for CRT to pass for that model of device.

NOTE This requirement means that the certifier cannot run some of the CRT tests on one device and others on an upgraded or otherwise modified device. It does not rule out a DUT which has a dual configuration.

6 Technical submissions from certification applicant

6.1 General

This clause defines requirements on the *technical submission* required from a certification applicant in order to support CRT. The technical submission consists of specific design information, end user documentation, hardware and software.

A subset of the design information required from the certification applicant is related to essential services supported by the device. Thus 6.2 begins with a discussion of the essential service concept. The hardware and software submission includes the device itself and other hardware and software to support the test environment.

6.2 Meaning of “essential service”

Conceptually, the set of essential services is a subset of device services that need to be available in order for the device to perform its intended function within a defined set of application environments. Thus for example a device must maintain the control loop and process view in all environments, and maintain certain history data for pharmaceutical environments. The impact of this concept on CRT is that to pass these tests, all services that are identified as being essential in the intended application environment(s) for a device per the requirements following, must be “adequately maintained” under network attacks and other adverse network conditions as simulated during testing.

ISCI identifies in this specification those services that are always considered essential, for all application environments, and therefore always subject to test. ISCI also identifies services that are essential in some application environments. An applicant for certification may explicitly “opt-out” of testing for the latter services. The certification report indicates whether or not the applicant opted out of testing for any of these services. If an opt-out is reported, this signifies to the end user that the device is not intended for use in application environments where those services not tested are essential.

NOTE ISCI has not defined a taxonomy mapping of application environments to required services. Thus the end user will determine whether a service that has not been tested for a certified device is essential for their environment.

Essential services fall into two classes, “upward” and “downward.” Downward services are the interface to the process being controlled. Upward essential services are interfaces to *peer or higher level* entities in the architecture. The following are the essential services in these classes for the purposes of CRT:

- Downward: *the process control/safety loop*
- Upward: *process view*, *command* (meaning change parameters of process control such as set points), *process alarms*. Other services of this type that are essential services unless explicitly excluded by the vendor are: *provide essential process history* and *peer-to-peer control communication*.

Thus essential services may be time-critical or non-time-critical services.

6.3 Testing parameters related to device essential services

Requirement CRT.R7 – Submission of essential service opt-outs

A certification applicant SHALL indicate whether or not they wish to opt-out of having the following services considered as essential services for the purposes of the CRT:

a) maintaining essential history data as described per Requirement CRT.R8;

b) maintaining peer to peer control communication.

Requirement CRT.R8 – Submission of definition of essential history data

A certification applicant that considers maintaining history data an essential service and does not opt-out of this per Requirement CRT.R7, SHALL describe those types of historical records and fields in these records that they consider to be essential history data.

For upward essential services, preparation for CRT as described in the following requirement, will require that the applicant for certification offer a well-defined method to determine if each such service is adequately maintained. This method will typically involve a higher layer supervisory component that communicates with their device. Thus for example, examination of a log might show that view was maintained, or was not provided by the device as expected. Under this approach higher layer system components become part of the “test harness” for the device. The CRT approach is to test an embedded device within its operational context, rather than standalone as a generic networked entity.

Requirement CRT.R9 – Submission of upward essential service monitoring criteria

The certification applicant and the test laboratory SHALL agree in advance of testing on documented test methods and criteria for monitoring device services. These methods MAY utilize existing higher level or peer systems designed to communicate with the device under test. These methods and criteria SHALL determine whether the following desirable operating conditions hold.

a) Whether the device is responding to commands to change parameters of the controlled process, in a timely fashion (such as change a set point);

b) Whether the device is providing a view of the process in a timely fashion;

c) Whether the device is providing process alarms in a timely fashion, including buffering alarms deemed undeliverable as designed;

d) Whether the device is maintaining essential history data, including buffering as designed essential history data deemed undeliverable, unless the applicant has opted-out of testing for this service;

e) Whether the device is providing timely peer to peer process communication, where this includes successfully sending messages over an unobstructed channel, and buffering as designed when such messages are deemed undeliverable, unless the applicant has opted out of testing for this service.

NOTE 1 The method for monitoring adequate maintenance of the control loop is not covered by this requirement since it is explicitly defined later in this specification.

Requirement CRT.R10 – Submission of method to achieve maximum recommended device load

A certification applicant SHALL submit a method that can be used to load the DUT to the maximum level recommended in documentation for end users, and a method to verify this load has been achieved and maintained.

NOTE 2 These methods will be used to load and verify the load on the device during robustness testing, since certification will require that essential services are maintained under all network traffic conditions as well as high load on the device.

NOTE 3 It is expected that different vendors will use different parameters or combinations of parameters to specify maximum load, for example CPU and memory utilization.

Requirement CRT.R11 – Submission of cycle time and control jitter tolerance

A certification applicant SHALL submit a time unit value for the cycle time of the embedded device; and a time unit value for maximum jitter tolerance for control output (value and confidence) that represent the expected performance of the embedded device.

NOTE 4 These values shall be used in determining pass/fail for CRT tests as described in Requirement CRT.R39.

6.4 Hardware/software

The requirements below apply to the embedded device hardware and software that is submitted for CRT, and other supporting hardware and software that may be needed.

Requirement CRT.R12 – Submission of device hardware and software

A certification applicant SHALL submit for CRT a product that is or will be unambiguously identifiable and specifiable by an end customer for procurement, in a hardware/software configuration that enables all of the procured software functionality of the product.

Requirement CRT.R13 – Submission of monitoring hardware and software for downward essential services

For a device that creates digital outputs using conveyance methods other than 0-5V, or analog outputs using other conveyance methods than 4-20mA, the certification applicant MAY be required by the test laboratory to submit a hardware and software test harness, that monitors whether the device adequately maintains control capability as defined by Requirement CRT.R39. The test harness SHALL be independent of the embedded device and provide a binary output based on this monitoring (maintained or not maintained) to the test device. In addition, for discrete device outputs, the certification applicant SHALL submit such a test harness.

NOTE A recognized CRT tool can directly monitor digital and analog outputs using the conveyance methods called out in this requirement, so in this case such a test harness would not be required. If a CRT tool has additional monitoring capability, then the test laboratory may not need to employ such a test harness.

Requirement CRT.R14 – Submission of monitoring hardware and software for upward essential services

A certification applicant SHALL submit to the certification process the hardware/software necessary to carry out device monitoring using the methods described in Requirement CRT.R9. Where the test laboratory already has the required equipment, they MAY waive all of part of this requirement at their discretion.

6.5 Device descriptive information

Requirement CRT.R15 – Submission of end user device documentation

A certification applicant SHALL submit to the certification process all documentation (printed, on-line or otherwise) that is delivered along with, or made available to, an end customer who purchases the product submitted for certification.

Requirement CRT.R16 – Submission of list of accessible network interfaces

A certification applicant SHALL submit to the certification process a list that clearly identifies all network interfaces present on the device that they define as *accessible* interfaces. The list of accessible interfaces SHOULD include all interfaces such that:

- the vendor recommends the interface to customers as suitable for use during normal operation; and
- the interface supports any protocol subject to CRT, for operation or instrumentation; and
- connection to the interface can occur without physical reconfiguration of the normal operational configuration.

Requirement CRT.R17 – Submission of implemented protocols

The certification applicant SHALL submit a list of all IP protocols that are supported on the device, for each accessible interface.

Requirement CRT.R18 – Submission of description of intended embedded device defensive behavior

For each protocol supported by the device which is covered by CRT, a certification applicant SHALL submit information that indicates one of:

a) traffic received under that protocol is not subject to rate limiting, in other words the design of the device does not distinguish between rates of incoming traffic

b) traffic received by the device is subject to rate limiting.

In case b) the applicant SHALL also provide a *known limited rate* which is a message or byte quantity per unit time which is known to be sufficient to ensure that the device will display its rate-limiting behavior, and SHALL describe the anticipated change in device behavior and the conditions under which behavior returns to “normal.”

Similarly, a certification applicant SHALL provide a description of any other defensive behavior employed by the device that may impact certification testing. For example the embedded device may employ IP address blacklisting, where an IP address is blocked if it previously has sent suspicious or excessive traffic to the device, or may employ a redundant configuration that provides automatic failover if one or more of the redundant units detects adverse conditions or fails.

NOTE Knowing a limiting rate in advance makes the test process more efficient, but the validity of the rate submitted will not impact pass/fail of CRT.

7 Common test reporting requirements

This clause contains requirements on test reporting that are common across all CRT tests. Additional requirements on reporting the results of interface surface testing and protocol-specific robustness testing are found in Clauses 8 and 9 respectively.

Requirement CRT.R19 – CRT report summary

The CRT process SHALL produce a summary report of all results of CRT testing, in addition to providing detailed test results.

Requirement CRT.R20 – Test report administrative information

The CRT process SHALL produce a test report that includes the following information:

- the vendor of the device under test;
- the applicant for the certification (typically the product vendor, but this may be another organization that owns the intellectual property associated with the device);

- the testing agent and contact information;
- an identifier that specifies the version of the software under test;
- an identifier of the ISASecure Test Specification version to which the testing conforms;
- the protocols tested, test suites employed and date(s) of testing; and
- date of the test report.

Requirement CRT.R21 – Report CRT test case descriptions

The CRT report SHALL include names for and high level descriptions of test cases executed. The required certification test suite is organized into meaningful test cases at the discretion of the test laboratory. However, the test laboratory SHALL make available to the device vendor, a mapping from their test cases to the tests enumerated in clause 7 of the ISASecure robustness testing specification for each tested protocol.

Requirement CRT.R22 – Report CRT methodology summary

The CRT report SHALL provide a high level summary of the methodology used to conduct each type of test.

Requirement CRT.R23 – Report CRT configuration

The CRT report SHALL describe the test configuration used to conduct the tests, including the configuration of the device under test.

Requirement CRT.R24 – Report ISASecure reference for test failure

For any test outcomes that result in a certification not being granted, the CRT report SHALL reference the applicable requirement(s) of the ISASecure test specification upon which that test is based.

Requirement CRT.R25 – Report test failure analysis

For any test failures, whether or not they result in a certification not being granted, the CRT report SHALL describe the discussion, analysis and conclusions reached regarding the failure that took place between the test laboratory and the applicant for certification.

Requirement CRT.R26 – Report conditional branches of test execution

The test report SHALL indicate whether any branches of testing were executed based upon test branching logic that was triggered by prior anomalous observed testing results.

Requirement CRT.R27 – Report test software version

The CRT report SHALL provide full version identifiers or hash values that, taken together with the test laboratory's procedures, unambiguously define the specific test software used to carry out all tests, to support reproducibility of test results.

Requirement CRT.R28 – Report test identification and parameters for reproducibility

The CRT report SHALL provide information sufficient to support the unambiguous reproducibility of the test, such as a test version and any parameters such as the pseudo random test seed used to generate network traffic for a test. Where applicable the report SHOULD provide a network trace of the traffic that preceded a test failure using a tool for packet capture.

8 Interface surface test

8.1 General

The interface surface test is the first test run as part of the CRT. It has two purposes:

- to determine the ports and services active on the DUT, which in turn determines which protocols will be subject to robustness testing under the CRT; and
- to test whether essential services on the device are adequately maintained during a port scan.

NOTE While the nmap tool (http://nmap.org) is typically used for these purposes, its use is not a requirement in this specification.

8.2 Test configurations

8.2.1 Basic interface surface test configuration

The basic interface surface test configuration consists of a TD sending packets to the DUT, augmented by components as required to monitor the performance of the DUT essential services. Two test configuration variants are used during the interface surface test, corresponding to the two purposes of the test. This is due to the fact that many embedded devices are deployed in conjunction with separate or integrated firewall functions. For the purposes of determining ports and services that are active on the DUT, where possible, any such filtering functions are configured to be disabled, so that the basic protocol capabilities of the underlying device can be more readily determined. For the purposes of examining how the device maintains essential services during a port scan, the firewall functions are configured as they would be by the end customer.

Requirement CRT.R29 – Basic interface surface test configuration

The configuration for the interface surface test SHALL include the following elements:

a) the device under test;

b) a testing device or devices that generates the network traffic/stimuli required to carry out the testing;

c) the configuration required to carry out the methods for monitoring upward essential services as described per Requirement CRT.R9;

d) the configuration described in Requirement CRT.R30 that is required to monitor downward essential services; and

e) a wired switched or non-switched network path that connects all of the above components.

NOTE Typically the system used to send the network traffic for the scanning process is the host for nmap or a similar tool.

8.2.2 Configuration for downward essential services monitoring

Requirement CRT.R30 – Configuration for downward essential services monitoring during interface surface test

The test configuration for interface surface testing SHALL include the following to allow monitoring of the control/safety loop:

a) a control program on the embedded device that provides an observable expected control output, specified in Requirement CRT.R39

b) control programs and any additional devices necessary to load the DUT to the predefined maximum level recommended to end users, and to verify this load has been achieved and maintained by the methods specified per Requirement CRT.R10

c) a testing device which is a monitoring component that is capable of receiving the digital control output of the DUT and collecting data to support the calculation of jitter on the signal received in ms, to the accuracy stated in Requirement CRT.R39.

NOTE 1 This testing device need not be the same physical device as the testing device that generates network traffic for the test.

NOTE 2 This testing device can calculate jitter in real time or the calculation can be done off-line based on data from the testing device. Performing the calculation in real time permits more advanced test branching based upon observed results.

8.2.3 Configuration for firewalls

Requirement CRT.R31 – Configuration for firewalls during interface surface test

The test configuration for the interface surface test SHALL support option a) as follows, and SHOULD support option b) where possible.

a) Any firewall available on the DUT and any firewalls intermediary between the DUT and the TD SHALL be preconditioned in the configuration defined for end customer use.

b) Any firewall available on the DUT and any firewalls intermediary between the DUT and the TD SHALL be preconditioned by disabling any rules that block the transmission of any type of network traffic.

8.3 Test procedure

The interface surface test procedure involves scanning for UDP and TCP ports that may be active and for IP protocol types.

A UDP TPDU addressed to a closed port SHOULD be replied to with an ICMP Port Unreachable PDU. However, since such a reply can itself be used as a multiplying factor in DoS attacks and as a means of gaining information about the queried subsystem, the DUT also MAY ignore the UDP TPDU and not generate such an ICMP reply PDU. Hence either of these results is interpreted to mean that the port is not active.

Requirement CRT.R32 – UDP port scan

The interface surface test SHALL include a scan of all (0-65535) DUT UDP ports to determine which of those ports is active. This scan SHALL be performed against all accessible interfaces per Requirement CRT.R16. The scan SHALL take the form of UDP TPDUs with non-zero (and preferably plausible) content sent to each of the 64 Ki possible UDP ports that the device may recognize. The device may respond to this testing with a Port Unreachable ICMP PDU per RFC1122, 4.1.3.1, or it may ignore the received UDP TPDU. Any other response SHALL imply that the port may be active. The test configuration for this scan SHOULD meet Requirement CRT.R31 b).

Requirement CRT.R33 – TCP port scan

The interface surface test SHALL include a scan of all (0-65535) DUT TCP ports to determine which of those ports is active. This scan SHALL be performed against all accessible interfaces per Requirement CRT.R16. The scan SHALL take the form of an attempt to establish a complete TCP connection sent to each of the 64 Ki possible TCP ports that the device may recognize. The device may respond to this testing with a Port Unreachable ICMP PDU, or it may ignore the connection attempt. Any other response SHALL imply that the port may be active. The test configuration for this scan SHOULD meet Requirement CRT.R31 b).

Example nmap commands that would achieve these last two requirements for an IPv4 device using the nmap tool version 5.21 are:

nmap –sU –vv –p0-65535 target_ip, for UDP

nmap –sT –vv –p0-65535 target_ip, for TCP

In these commands the parameters have meanings as follows:

- -sU designates a UDP scan
- -sT designates a TCP connection scan, which is distinct from a SYN scan (-sS) in which a complete TCP connection is not established
- -vv requests “very verbose” feedback from nmap while the scan is progressing
- -p0-65535 designates that all possible ports should be scanned
- Target_ip the IP address for the DUT

Requirement CRT.R34 – Use of DUT-based utilities for determining active ports

If a utility is available that runs on the DUT with the capability to identify open ports, this utility SHALL be run as part of the interface surface test and used to augment the results of the port scans. Such local utilities SHALL NOT be used as the sole source of data in this matter.

NOTE 1 Such a utility would be analogous to “lsof” or “netstat” which run on Unix based systems.

Requirement CRT.R35 – IP protocol type scan

The interface surface test SHALL include a scan for all IP protocol types. The test configuration for this scan SHOULD be per Requirement CRT.R31 b).

The next requirement takes into account the fact that some embedded devices may have several accessible network interfaces, and may be placed in one of several operating modes such as control mode, configuration mode, or update mode.

Requirement CRT.R36 – Scan coverage of all accessible network interfaces and device modes

If the DUT supports several modes of operation in which different device functions are available in different modes, the interface surface test SHALL include a UDP port scan, TCP port scan, and IP protocol type scan of the DUT in all of these modes, over each accessible network interface, while running all essential services that are available in these modes in such a way as to support monitoring of upward and downward essential services.

Requirement CRT.R37 – High rate port and protocol scans

The interface surface test SHALL include a two-phase test case in which the UDP port scan, TCP port scan and IP protocol type scan described in Requirement CRT.R32, Requirement CRT.R33 and Requirement CRT.R35 are each performed repeatedly. In the first phase, the repetition occurs at a high rate, but less than that the rate at which rate limiting occurs as declared by the device vendor. In the second phase, the repetition occurs at a rate up to the auto-negotiated maximum rate of the underlying network, maintains the high load rate for a few seconds, and then gradually reduces its sending rate to zero.

Requirement CRT.R38 – Reproducibility of determination of ports that may be active

The method for determining which UDP and TCP ports may be active SHALL be reproducible.

NOTE 2 For example, if using nmap, one would record the version of nmap used for the scan.

8.4 Test pass criteria

8.4.1 General

The set of potentially active UDP or TCP ports and/or other IP-based protocols does not determine whether a DUT passes the interface surface test. Pass/fail is determined by the behavior of the DUT during the port scans that comprise this test.

In the interface surface test, the device is subjected to a variety of scans at different rates. In overview, an embedded device will pass the interface surface test if it adequately maintains essential services throughout the tests. Clause 6 defines those services that are essential.

This clause defines what it means to adequately maintain essential services. The general definitions are provided in 8.4.2, followed by requirements in 8.4.3 that specify how these definitions are applied in the context of the CRT.

8.4.2 Meaning of “adequately maintain essential services”

8.4.2.1 General

The meaning of the term “adequately maintain an essential service,” is dependent upon the particular essential service. It is defined as follows for each such service. Among all downward services and all upward services, these definitions are nearly identical. In summary, it is acceptable for upward services to be lost due to interference from flooding on their own network interface, but not due to any other network traffic conditions. It is not acceptable for downward services to be lost under any network traffic conditions.

8.4.2.2 Meaning of “adequately maintain control capability”

An embedded device is said to adequately maintain control capability if the control loop (safety loop) is maintained with the existing control parameters, under any network traffic conditions on all accessible device network interfaces. This definition describes the opposite of “loss of control.” Note this is distinct from the capability to command a change to the parameters controlling the process, described next.

8.4.2.3 Meaning of “adequately maintain command capability”

An embedded device is said to adequately maintain command capability if network traffic on the interface used for commanding the device does not disable the capability to respond to commands from higher level systems in a timely fashion, other than via continuous flooding. A device may intentionally disable command response temporarily due to invoking a defense mechanism against flooding. However if it invokes such a defense, it shall return to normal processing once flooding ceases, in a manner consistent with the documented design. Network traffic on other accessible device interfaces shall not interfere with the capability to achieve timely response to commands.

8.4.2.4 Meaning of “adequately maintain view”

An embedded device is said to adequately maintain view if network traffic on the interface used for providing process view cannot disable the capability to provide this view in a timely fashion, other than via continuous flooding. A device may intentionally disable process view temporarily due to invoking a defense mechanism against flooding. However if it invokes such a defense, it shall return to normal processing once flooding ceases, in a manner consistent with the documented design. Network traffic on other accessible device interfaces shall not interfere with the capability to provide a process view in a timely fashion. This definition describes the opposite of “loss of view.”

8.4.2.5 Meaning of “adequately maintain alarms and alarm reporting”

An embedded device is said to adequately maintain alarms and alarm reporting if network traffic on the interface used for sending process alarms cannot disable the capability to send these alarms in a timely fashion, other than via continuous flooding. A device may intentionally disable alarm reporting temporarily due to invoking a defense mechanism against flooding. However if it invokes such a defense, it shall return to normal processing once flooding ceases, in a manner consistent with the documented design. Alarms shall be buffered while reporting is disabled, at the device-specific capacity. Network traffic on other accessible device interfaces shall not interfere with the capability for timely reporting of alarms.

8.4.2.6 Meaning of “adequately maintain essential history reporting”

An embedded device is said to adequately maintain essential history reporting if network traffic on the interface used for essential history reporting cannot disable the capability to send essential history data in a timely fashion, other than via continuous flooding. A device may intentionally disable history reporting temporarily due to invoking a defense mechanism against flooding. However if it invokes such a defense, it shall return to normal processing once flooding ceases, in a manner consistent with the documented design. Essential history data shall be buffered while reporting is disabled, at the device-specific capacity. Network traffic on other accessible device interfaces shall not interfere with the capability to achieve the timely reporting of essential history data.

8.4.2.7 Meaning of "adequately maintain peer-to-peer control communication"

An embedded device is said to adequately maintain peer-to-peer control communication if network traffic on the interface used for peer-to-peer control communication cannot disable the capability to send this communication in a timely fashion, other than via continuous flooding. . A device may intentionally disable peer-to-peer control communication temporarily due to invoking a defense mechanism against flooding. However if it invokes such a defense, it shall return to normal processing once flooding ceases, in a manner consistent with the documented design. Pending messages are buffered while communication is disabled, at the device-specific capacity. Network traffic on other accessible device interfaces shall not interfere with the capability to achieve timely control communication with peers. It is assumed that other than flooding of the device interface used for peer-to-peer control communication, the channel to the peer is unobstructed.

8.4.3 Criteria for "adequately maintain essential services"

The next two requirements following describe how the preceding definitions for "adequately maintain an essential service" are applied in the context of CRT. The final requirement relies on these requirements to provide criteria for passing the interface surface test.

Requirement CRT.R39 – Test criteria for "adequately maintain control capability"

An embedded device SHALL be determined to have adequately maintained control capability during a test if a specified cyclically-repeated waveform is measured to have a maximum observed time jitter over the test period that is less than the sum of the vendor's declared tolerance value per Requirement CRT.R11, plus maximum measurement jitter.

a) for devices that can create an analog output, each cycle of the waveform SHALL consist of 10 equal steps of increasing value and then 20 equal steps of decreasing value, both at one step per second, transitioning between the nominal minimum and maximum values of the output device;

b) for devices that can create a digital output, the waveform SHALL consist of a rectangular wave with a 2/3 duty cycle and 3 s period, of 1 s at nominal "1" and 2 s at nominal "0"; and

c) these waveforms SHALL be generated by the ladder/control/supervisory logic of the device, and not autonomously by the I/O logic

d) both digital and analog outputs with these characteristics SHALL be measured if both are present

e) if digital or analog outputs can be conveyed using more than one method (such as via pneumatic, electrical, or using a Fieldbus message), then these outputs for all supported forms of conveyance SHALL be monitored per the criteria of this requirement

f) any discrete outputs for all supported forms of conveyance SHALL be monitored using the test harness described in Requirement CRT.R13.

NOTE 1 This requirement is intended to permit the output monitoring process to detect anomalous behavior of the control software of the device, which monitoring could be defeated if low-level I/O were generating the waveform autonomously.

NOTE 2 The intent of this requirement is to test whether the supervisory logic continues to perform under adverse network conditions; it is not the intent of these tests to provide validation of the supervisory logic itself.

The jitter requirements of a) and b) are with respect to the relative timing of the transitions, not the analog value of the analog or digital output. The TD employed to test an embedded device shall itself introduce a maximum measurement error of no more than 10% of the cycle time of the DUT.

Requirement CRT.R40 – Test criteria for "adequately maintain upward essential services"

An embedded device SHALL be determined to have adequately maintained upward essential services during a test if it meets the definitions in 8.4.2.3 through 8.4.2.7, where the test criteria for determining the status of services as referenced in those definitions, are as agreed between the test laboratory and the vendor, per Requirement CRT.R9.

Requirement CRT.R41 – Criteria for "pass interface surface test"

The DUT SHALL pass the interface surface test if it adequately maintains all essential services (per Requirement CRT.R39 and Requirement CRT.R40), throughout all of the UDP and TCP port scans and IP protocol type scans performed to meet the interface surface test requirements.

8.5 Reproducibility criteria

Requirement CRT.R42 – Reproducibility of interface surface test failure

If the DUT fails to adequately maintain an essential service during a scan that is part of the interface surface test, this behavior SHALL be shown to be reproducible before the test is given a failed status.

8.6 Test report

A test laboratory performing interface surface testing provides a test report that meets the following requirements. In addition, all CRT test reports meet the common reporting requirements in Clause 7.

Requirement CRT.R43 – Report basic interface surface test information

The report for the interface surface test shall meet all basic test reporting requirements in Clause 7: Requirement CRT.R20 through Requirement CRT.R28, inclusive.

Requirement CRT.R44 – Report UDP ports that may be active

The list of UDP ports determined to not be either ignored or unreachable for each accessible device interface and mode, and associated protocols per [PORT], SHALL be included in the interface surface test report.

Requirement CRT.R45 – Report TCP ports that may be active

The list of TCP ports determined to not be either ignored or unreachable for each accessible device interface and mode, and associated protocols per [PORT], SHALL be included in the interface surface test report.

Requirement CRT.R46 – Report IP protocol types

The list of IP protocol types that appear to be supported for each accessible device interface and mode SHALL be included in the interface surface test report.

NOTE The appropriateness of potentially active UDP and TCP ports and supported IP protocol types is examined in the functional security assessment. Additionally, the UDP and TCP port information partially determines the scope of protocol-specific robustness testing.

Requirement CRT.R47 – Report behavior of essential services during scans

For each essential service identified for the DUT per Clause 6, the interface surface test report SHALL state whether the service was adequately maintained (per the definition in Requirement CRT.R39 and Requirement CRT.R40) during the port scans that comprise the interface surface test, and if not, describe its behavior, the network interface used and the device mode if applicable.

9 Protocol-specific robustness tests

9.1 General

The interface surface test described in Clause 8, taken together with information provided by the certification applicant, determines for a given device, those protocols that are subject to protocol-specific robustness testing in the CRT (as stated in requirements Requirement CRT.R1 and Requirement CRT.R2). Requirements for testing each protocol are defined in individual specifications per protocol, which are listed in [EDSA-300]. However, test requirements that are common across all protocols are defined in this

specification. Individual protocol robustness test specifications refer to this specification for those requirements.

9.2 Test configuration

9.2.1 General

This clause describes test configuration requirements that apply to protocol-specific robustness testing, across classes of protocols. This includes network, DT, and DUT configuration as well as configuration related to monitoring essential services. The individual protocol specifications will state which test configuration as described here applies to testing that protocol.

These requirements describe two test configurations, one of which requires a non-switched connection between the TD and the DUT. A non-switched network connection between the TD and the DUT is required for some tests because intervening switching equipment may in fact correct the very protocol errors that the TD is attempting to send to the DUT for low level protocols such as ARP and Ethernet II.

As specified by the following test configurations, the intent for CRT is that an embedded device is tested in the environment within which it is used, which (since it is an embedded device) includes network connections to higher level supervisory components and to entities that receive control signals.

9.2.2 Test configuration 1

Requirement CRT.R48 – Test configuration 1: Switched IP connection from TD to DUT

The test laboratory SHALL support a test configuration for protocol-specific robustness testing that has the following elements:

a) the device under test;

b) a testing device or devices that generate network traffic required to carry out the testing;

c) the configuration required to carry out the methods for monitoring upward essential services as described per Requirement CRT.R9;

d) the configuration as described in Requirement CRT.R30 that is required to monitor downward essential services;

e) a wired switched or non-switched network path that connects all of the above components.

NOTE 1 This is the same as the configuration for the interface surface test, except that the TD in this case will be generating packets for specific protocols rather than the network scans used for the interface surface test.

NOTE 2 Either a switched or non-switched connection will support the tests that will use this configuration. This is because those tests do not generate erroneous traffic that a switch will discard. Hence this choice may be a matter of convenience and is left to the discretion of the tester.

9.2.3 Test configuration 2

Requirement CRT.R49 – Test configuration 2: Non-switched IP connection from TD to DUT

The test laboratory SHALL support a test configuration for protocol-specific robustness testing that has the following elements:

a) the device under test;

b) a testing device or devices that generate network traffic required to carry out the testing

c) the configuration required to carry out the methods for monitoring upward essential services as described per Requirement CRT.R9;

d) the configuration described in Requirement CRT.R30 that is required to monitor downward essential services;

e) a network path that connects the TD and the DUT, such that intervening network components do not interfere with traffic on this path, whether or not it is erroneous;

f) a switched or non-switched network path that connects all other pairs of the above components.

NOTE Test configuration 2 differs from Test configuration 1 only in that there cannot be an intelligent hub or a switch between the TD and the DUT in test configuration 2, since it would likely drop erroneous traffic that is required to run some tests.

9.3 Test procedure

9.3.1 General test approach

The following requirements describe protocol-specific robustness testing that takes place following the interface surface test described in clause 8.

Requirement CRT.R50 – Robustness testing phases

Robustness testing SHALL consist of three conceptual phases, where the last two may overlap.

a) The first conceptual phase, Baseline operation, attempts to demonstrate that the selected DUT protocol suite used for testing appears to operate properly for simple test cases under low load, before any protocol fuzzing or stress testing is attempted.

   NOTE 1 This initial demonstration of apparently correct behavior establishes the presumption that failure during additional testing is due to vulnerabilities of the specific protocol under test, rather than other protocols in the test suite.

b) The second conceptual phase, Basic robustness testing, probes the implementation of each discovered feature for sensitivity to boundary conditions and special cases, and where at most one field in a message at a time is erroneous, as defined in 9.3.2. Basic robustness testing SHALL cover all fields in messages for the protocol under test and SHALL include sending sequences of erroneous messages to uncover any cumulative impacts on the device operation.

   NOTE 2 This conceptual phase focuses on simple protocol robustness/fuzzing tests.

c) The third conceptual phase, Load stress testing, probes the implementation's response to high traffic rates incorporating valid PDUs.

   NOTE 3 This conceptual phase focuses on load/performance tests, first under high but supposedly sustainable receiver load, then under massive overload.

Although the robustness testing of this specification is conceptualized as occurring in distinct logical phases that progress from simple single-factor testing to more complex load testing, there is no requirement that an actual robustness test process work in this ordered, sequential manner.

NOTE 4 The last sentence of Requirement CRT.R50b) implies that any tests that create errors in several fields of the same message cannot be counted toward the requirements of basic robustness testing to probe each individual field. This is because correct handling of a message with several errors does not imply that each of the errors occurring individually would be handled correctly. Tests with multiple errors per message are permitted but not required by this specification.

NOTE 5 Detailed requirements for these phases that are unique to each protocol are defined in the ISASecure CRT specifications for individual protocols that are listed in [EDSA-300].

Requirement CRT.R51 – Test coverage for devices with redundant configurations

If the DUT has a redundant configuration, then basic and load stress robustness testing SHALL be applied to the device when one or more of the redundant units are not operational. In particular, these tests SHALL cover each possible number of operational/non-operational units for which the overall device is designed to remain operational.

Later in this document are requirements on special field values that should always be used during robustness tests. The following requirement provides a more general coverage requirement.

Requirement CRT.R52 – Test coverage of field values

Basic robustness testing SHALL adhere to a well defined approach for providing overall coverage of protocol field values by generated traffic.

NOTE A simple example of such an approach is to require 2^n values for an n bit field.

Some embedded devices implement defensive mechanisms such that the device will refuse traffic from an IP address that has previously sent it suspicious, possibly erroneous or excessive traffic. This can make it difficult to run robustness tests for such a device if the capability cannot be turned off. The following requirement addresses the need to test such devices.

Requirement CRT.R53 – Robustness testing with IP address blacklisting

Robustness tests SHALL have the capability to generate various source IP addresses in traffic created for robustness testing, in order to successfully test embedded devices that employ IP address blacklisting.

Requirement CRT.R54 – TD traffic rate

A TD shall be capable of sustaining, at a minimum, a traffic rate of 100 Mbps and 250k IP packets per second.

9.3.2 Terminology

The terms “malformed”, “invalid values”, “contextually inappropriate” and “erroneous” are used in various protocol-specific CRT specifications. Each has a distinct and specific meaning:

- **Malformed**: A PDU or PDU field is *malformed* when it is structurally incorrect in one or more of the following ways (which also may overlap in some situations):
  - — is composed of a non-permitted sequence of protocol subfields,
  - — consists of an impermissible truncation of a permitted sequence of subfields,
  - — contains a field truncated to less than the permitted minimum length,
  - — contains a field extended beyond the maximum permitted length
  - — contains a field which has an actual length different than that specified elsewhere in the PDU or previously established,
  - — contains fewer than the permitted minimum number of repetitions of a subfield,
  - — contains more than the permitted maximum number of repetitions of a subfield, or
  - — contains a different number of repetitions of a subfield than that specified elsewhere in the PDU or previously established.
- **Invalid values**: A PDU field contains an *invalid value*
  - — when the specific field value is not among the statically defined permitted values for the field, or
  - — when, due to context, the value is not among the dynamically permitted values for the field.
- **Contextually inappropriate**: A PDU, or a PDU field such as a PDU option field, is *contextually inappropriate* when, under the governing rules of the protocol, the PDU or field should not occur in the sequence of PDUs or PDU fields where it is found. Examples include repetitions of a PDU option field that is permitted only once in any specific PDU, the presence of two mutually-incompatible PDU option fields in the same PDU, or the occurrence of a reply PDU at a time other than when a prior request PDU has authorized the reply.

  NOTE Some dynamically invalid values are also contextually inappropriate.

- **Erroneous** is the broadest term used in the CRT specifications. A PDU, or a field value within a PDU, is *erroneous* when it is malformed, contains invalid values, is contextually inappropriate, or contains contextually inappropriate values.

The requirements in 9.3.3 and 9.3.4 list specific types of both erroneous and valid values for fields, that robustness testing will cover. Use of these values is required unless some unusual circumstance precludes using them as part of a practical test. An example of such a circumstance is a case in which sending a particular value triggers an intended defensive behavior that locks up the device for some period of time.

9.3.3 Fields of simple type

9.3.3.1 Fixed-length fields representing integers or enumerations

Requirement CRT.R55 – Required test values used in testing fixed-length fields representing integers or enumerations

The set of tested values SHALL include both the two endpoint values and the two median values of the underlying representation, as well as representable adjacent values.

NOTE 1 Some of these values may be invalid or erroneous in the test PDUs.

NOTE 2 When the integer values or enumerations use a signed 2's-complement representation of N bits, the range of the underlying representation is $[-2^{-(N-1)} .. +2^{(N-1)}-1]$, the two endpoint values are $-2^{-(N-1)}$ and $+2^{(N-1)}-1$, the two median values are –1 and 0, and the representable adjacent values are $-2^{-(N-1)}+1$, -2, +1 and $+2^{(N-1)}-2$.

NOTE 3 When the integer values or enumerations use an unsigned representation of N bits, the range of the underlying representation is $[0 .. 2^N-1]$, the two endpoint values are 0 and 2^N-1, the two median values are $2^{(N-1)}-1$ and $2^{(N-1)}$, and the representable adjacent values are 1, $2^{(N-1)}-2$, $2^{(N-1)}+1$ and 2^N-2.

When the set of valid values for a field has limits [P .. Q] other than those of the underlying representation, then the set of tested values also SHOULD include both the two endpoint limit values P and Q as well as representable adjacent values.

NOTE 4 Those latter values are P–1, P+1, Q–1 and Q+1 when they fall within the span of the underlying representation.

9.3.3.2 Determined-length fields containing varying-length self-delimiting strings

Requirement CRT.R56 – Required test values used in testing determined-length fields containing varying-length self-delimiting strings

In some cases a field whose amount of allocated storage (i.e., maximum length in bytes) is determined by other means, either in the underlying protocol specification or by a separate length field or indicator, contains a self-delimiting string, typically terminated by a zero (null) value. In such cases testing of string values SHALL include

a) the null string (i.e., where the first string element in the field is the terminating element, typically zero), and

b) a string that occupies the entire allocated storage and that does not include the terminating element.

In cases where the string contains multi-byte characters, such as 16-bit Unicode, testing also SHOULD include

c) a string that apparently terminates in the middle of a multi-byte character, to the extent that such coding violations are possible.

When the coding of multi-byte characters uses an escape mechanism so that all characters are not identical length, then testing also SHALL include

d) a string that is a combination of b) and c), i.e., that exhausts the string storage within the last character, which is not a valid terminating element.

In cases where the terminating element of a string is itself a multi-byte character or character sequence, i.e., a <CR><LF> sequence at the end of each line of terminal input, testing also SHALL include

e) a case related to cases b) and c) where the last character is the beginning of that valid multi-byte (or multi-character) terminating element, but the complete element is not present.

9.3.4 Fields with substructure

9.3.4.1 Fields with a varying sequence of fixed-size subfields

Requirement CRT.R57 – Testing fields with a varying sequence of fixed-size subfields

When a field is defined to contain a varying sequence of fixed-size subfields, the sequence of constructed test PDUs SHALL include

- PDUs that conform to the sequence rules imposed by the protocol, in terms of type, order, number of repetitions of subfields and total number of subfields, and
- PDUs that violate one or more of the sequence rules imposed by the protocol.

In particular, PDUs that contain too few fields, too many fields, and incorrect sequences all SHALL be included in the test.

9.3.4.2 Fields of self-defining length

Requirement CRT.R58 – Testing fields with substructure and self-defining length

Many PDU option fields, and some PDU fields that convey one or more self-delimiting strings, have a length that is determinable only by scanning and parsing the contained subfields. In such cases testing SHALL include fields encoded such that such parsing leads to erroneous results. Where applicable, testing SHALL include nesting of substructures, and the use of previously established testing processes for all strings within substructures which contain delimiters and termination characters.

9.3.5 Guidance on protocol-specific load stress testing

Requirement CRT.R59 – Protocol-specific load testing

Load testing SHALL include.the TD sending a flurry of valid PDUs to the DUT just below the vendor-disclosed rate limit (if the device is rate limiting) and at the full auto-negotiated link rate.

9.4 Test pass criteria

In the protocol-specific robustness tests, the device is subjected to a variety of protocol errors and network traffic rates. In overview, an embedded device will pass protocol-specific robustness testing if it adequately maintains essential services throughout the tests. Clause 6 defines those services that are essential.

Requirement CRT.R60 – Criteria for protocol specific robustness test pass

An embedded device shall pass the communication robustness test for a specific protocol if:

a) it adequately maintains all upward and downward essential services throughout the test, as defined in Requirement CRT.R39 and Requirement CRT.R40;

b) it meets other pass criteria, if any, that are explicitly stated in the CRT specification for that protocol.

9.5 Reproducibility criteria

Test reproducibility assists both test laboratory personnel and vendors in demonstrating unexpected test results and identifying their causes through repetition of the testing, often after enabling instrumentation within or applied to the software under test.

Requirement CRT.R61 – Reproducibility of protocol-specific robustness test failure

If the DUT fails to adequately maintain an essential service or exhibits other behavior that indicates a failure during a protocol-specific robustness test, this behavior SHALL be reproducible before the test is given a failed status.

Requirement CRT.R62 – Generation of reproducible robustness tests

A documented reproducible deterministic process (which MAY be a seeded pseudo-random process) SHALL drive basic robustness and load stress testing, so that, for a specific implementation of the software under test, these phases of the testing comprise a 100% reproducible process with no variance..

Requirement CRT.R63 – Pseudo-random seed value

When a pseudo-random test process is employed to drive robustness testing, that process SHALL be keyed by an initial seed value of at least 16 bits that SHALL be listed in the test report.

Requirement CRT.R64 – Pseudo random seed reuse

When a pseudo-random test process is employed, the test framework SHALL provide a mechanism whereby a prior seed value can be configured before the test starts, so that a re-test of unchanged software will generate the identical test sequence provided that the response sequence is unchanged.

It is the responsibility of the implementer of the protocol software in the device under test (DUT) to ensure reproducibility of responses to a duplicated test sequence; when that is not possible, the assistance in diagnosis that observable retest can provide may not be available.

The test process MAY use feedback during the test selection process, so that detected apparent anomalies in the responses of the DUT can trigger focusing of subsequent tests on those protocol aspects that appear to cause the anomalies to manifest. Thus any change or correction in the software under test may cause the attack resistance tests to diverge at points where those changes affect the DUT's responses, thereby limiting the extent to which instrumented retest can assist in diagnosis of the cause of the discovered anomalies.

9.6 Test report

9.6.1 General

A test laboratory performing protocol-specific robustness testing provides a test report that meets the following requirements. These reporting requirements are common to all protocols. Any additional reporting requirements unique to a specific protocol are discussed in the ISASecure robustness test specification for that protocol. In addition, all CRT test reports meet the common reporting requirements in Clause 7.

Requirement CRT.R65 – Report basic protocol specific robustness test information

The report for a protocol-specific robustness test shall meet all of the basic test reporting requirements in Clause 7: Requirement CRT.R20 through Requirement CRT.R28, inclusive.

Requirement CRT.R66 – Robustness results summary over all protocols

The protocol-specific robustness test report SHALL include a summary section that provides a high level overview of results covering all protocols tested for robustness.

9.6.2 Robustness phase report

These requirements relate to reporting on the basic and load stress phases of the robustness testing.

Requirement CRT.R67 – Report robustness failures

The protocol-specific robustness test report SHALL document any robustness test cases under which there were observed failures, where pass/fail criteria are defined in 9.4.

Requirement CRT.R68 – Report robustness failure conditions

For robustness tests which had an observed failure, the protocol-specific robustness test report SHALL document the test conditions that were associated with the failure.

Requirement CRT.R69 – Report robustness test case results listing

The protocol-specific robustness test report SHALL provide a listing of each category of robustness test cases executed, pass/fail status, a summary of any anomalous behavior observed for those test cases, and any related recommendations.

— — — — — — —