EDSA-405
ISA Security Compliance Institute – Embedded Device Security Assurance –
Testing the robustness of implementations of the IETF UDP transport protocol over IPv4 or IPv6

Version 2.6

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
2.1	2010.06.15	Initial version published to http://www.ISASecure.org
2.6	2010.09.17	Added test UDP.T08, probe of unknown application protocols at open UDP ports create distinct test criteria at high but supported rate and full auto-negotiated link rate; removed protocol conformance aspects of tests since covered by other industry efforts; removed discovery phase since not required to perform uniform testing over all devices; removed 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 robustness test specifications for embedded devices. The full current list of documents related to embedded device security assurance can be found on the web site of the ISA Security Compliance Institute, http://www.ISASecure.org.

1 Scope

This document is intended to provide requirements for testing the robustness of embedded device implementations of the IETF UDP protocol, 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. 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.

NOTE 1 The UDP protocol is stateless, without distinction between server and client roles.

NOTE 2 Although conformance is explicitly NOT a goal of this testing, prior versions of this document included some aspects of conformance testing which have now intentionally been removed.

2 Normative references

This associated specification contains requirements common to this and similar robustness tests for other protocols for embedded devices, including requirements on test configurations.

[EDSA-310] *ISA Security Compliance Institute – Embedded device security assurance – Common requirements for communication robustness testing of IP-based protocol implementations*[1], as specified at http://www.ISASecure.org

NOTE 1 Within this document, references to specific subclauses of this normative reference are made through symbolic tags of the form [CRT.Symbolic_tag]; the resolution of those tags is made in [EDSA-310], Table 1.

These publications of the Internet Engineering Task Force (IETF) are the controlling specifications for the protocol whose robustness testing is the subject of this document:

NOTE 2 For each RFC*nnn*, the controlling version can be found at http://tools.ietf.org/html/rfc*nnn*.

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

RFC768, *User datagram protocol*

RFC1122, *Requirements for internet hosts – communication layers*

NOTE 3 Only 4.1 is referenced.

RFC2460, *Internet protocol, version 6 (IPv6)*

NOTE 4 Only 8.1 is referenced.

NOTE 5 Other IETF specifications related to the above can be found in the Bibliography.

[1] to be published concurrently with this document

3 Definitions and abbreviations

3.1 Definitions

3.1.1
device under test
device that is being stimulated and observed during testing to demonstrate the characteristics and behavior of the device when presented with the selected sequence of test stimuli

3.1.2
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 [CRT.Terminology_of_Erroneous].

3.1.3
"Ethernet"
either the IETF Ethernet II protocol or IEEE 802 SNAP over IEEE 802.2 Type 1 LLC over IEEE 802.3

3.1.4
fragmenting
function performed by IPv4 to map one unfragmented NPDU into multiple smaller fragmented NPDUs before transmission

NOTE The equivalent OSI terms is segmenting, as specified in ISO/IEC 7498 1:1994, 5.8.1.9.

3.1.5
inferior (protocol)
protocol at a lower layer or sublayer than the referenced protocol

3.1.6
lower tester
tester that controls and observes a protocol layer implementation in a DUT through stimulus and observation via lower protocol layers and a physical interconnection to the TD

NOTE This is the only type of testing used in the ISCI EDSA robustness tests.

3.1.7
malformed (message or PDU)
PDU that violates syntactic rules on PDU structure

NOTE This is addressed further in [CRT.Terminology_of_Erroneous].

3.1.8
reassembling
post-reception function performed by IP to reconstruct one unfragmented NPDU from multiple fragmented NPDUs

3.1.9
superior (protocol)
protocol at a higher layer or sublayer than the referenced protocol

3.1.10
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

NOTE This could be any programmable network-connected device capable of processing PDUs at the rate required for testing.

3.1.11
upper tester
tester that controls and observes a protocol layer implementation in a DUT through stimulus and observation via a DUT-internal service interface between test software and the protocol layer under test

3.1.12
vulnerability
flaw or weakness in a system's design, implementation, operation, or management that could be exploited to violate the system's integrity or security policy

3.2 Abbreviations

The following abbreviations are used in this document

APDU	application-layer protocol data unit
CRT	communication robustness testing
DPDU	data-link-layer protocol data unit
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
IETF	Internet engineering task force
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)
(*N*)PDU	(*N*-layer) protocol data unit, where *N* = D (data-link), N (network), T (transport), A (application), etc
NPDU	network-layer protocol data unit
SNAP	sub-network access protocol
TD	testing device
TPDU	transmission-layer protocol data unit
UDP	user datagram protocol

4 Elements of the protocol under test

4.1 General

This document specifies robustness testing for the IETF UDP protocol, which is a stateless transport protocol providing an unordered, unprioritizable, unreliable end-to-end communications path.

4.2 UDP TPDUs

4.2.1 UDP TPDU structure

A UDP TPDU is structured as shown in Figure 1, using a big-endian octet order.

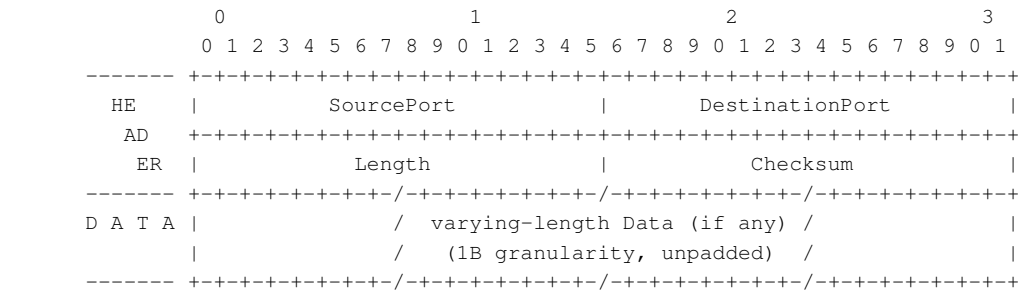

Figure 1 – UDP TPDU structure

4.2.2 Mandatory fields

The following fields are mandatory components of each UDP TPDU (where field sizes are specified in octets (B) or bits (b)):

a) SourcePort: (2 B): source UDP-TSAP-identifier; default 0x0000. See IANA port numbers

b) DestinationPort (2 B): destination UDP-TSAP-identifier. See IANA port numbers

c) Length (2 B): UDP TPDU length in octets, as an unsigned number

NOTE This should have a value of eight or greater.

d) Checksum (2 B): unkeyed message integrity code. See 4.2.3.2

e) Data (<Length -8> B, possibly null, granularity 1 B, not padded to a multi-octet boundary)

4.2.3 Mandatory protocol aspects

4.2.3.1 Conveying IP NPDU

UDP fixes the value of one of the header fields of any conveying IP NPDU.

- For IPv4, the NPDU header SHALL specify the UDP protocol in its ProtocolType field:
  - ProtocolType: 0x11 (UDP)
- For IPv6, the last header in the NPDU SHALL specify the UDP protocol in its NextHeader field:
  - NextHeader: 0x11 (UDP)

4.2.3.2 Checksum procedure

The checksum field is the 16-bit one's complement of the one's-complement sum of all 2 B words, in big-endian octet order, in a virtual UDP TPDU created by prefixing the actual TPDU by a pseudo-header consisting of extra 4 B words. The pseudo-header for use with IPv4 is shown in Figure 2; the pseudo-header for use with IPv6 is shown in Figure 3.

Each pseudo-header contains the Source IP Address; the Destination IP Address; the IP protocol ID for UDP (0x11), which in IPv6 is used as the value of the NextHeader field of the last header in the IPv6 NPDU; and (usually redundant with that of the UDP header itself) the length in octets of the conveyed UDP TPDU. Together these give the UDP TPDU some protection against being reconstructed from misrouted fragmented NPDUs.

NOTE This pseudo-header information is carried in the IP NPDU and is transferred across the Transport/Network layer service interface.

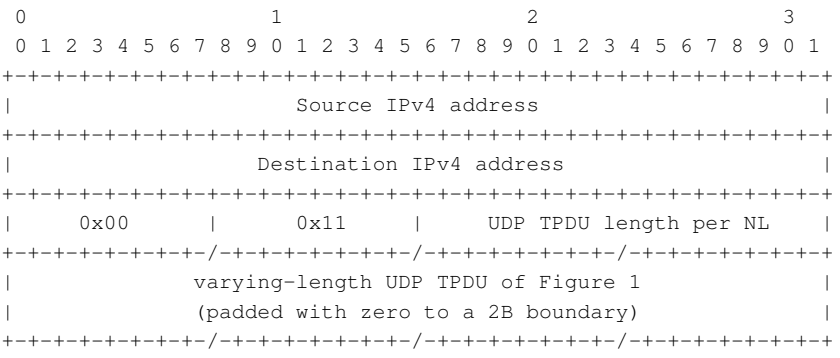

Figure 2 – Virtual UDP over IPv4 TPDU used for checksum computation

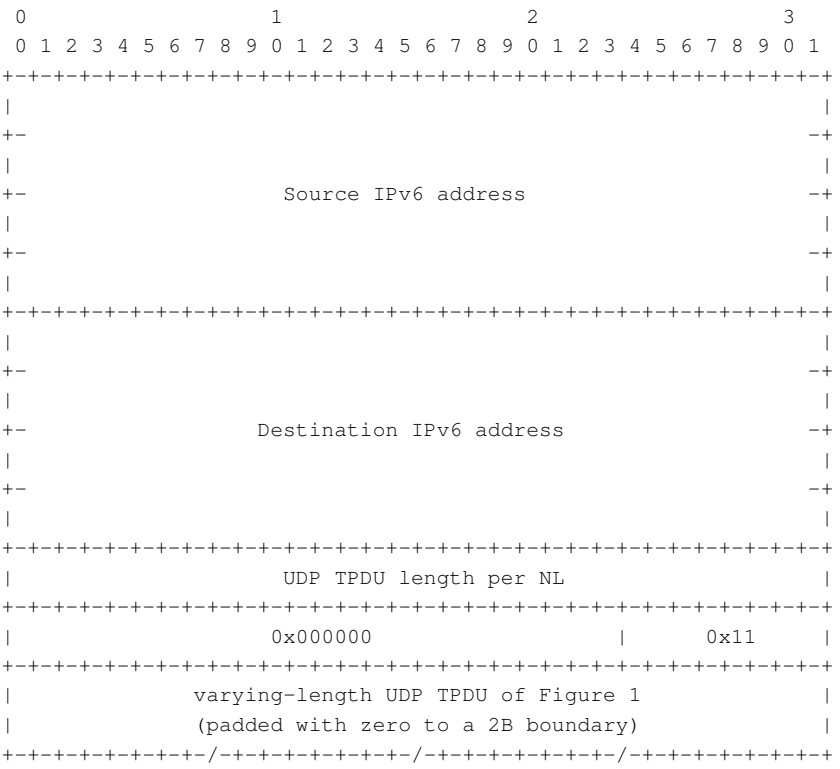

Figure 3 – Virtual UDP over IPv6 TPDU used for checksum computation

If a UDP TPDU contains an odd number of octets to be checksummed, an octet of zeros is appended when forming the virtual UDP TPDU that is used for checksum computation, to ensure an integral number of 2B words, as shown in Figure 2 and Figure 3. Within the virtual UDP TPDU, while computing the checksum, the checksum field of the contained varying-length UDP TPDU is set to zero.

The UDP TPDU length in the third 4B word of the virtual header is the length of the TPDU as presented to or reported by the associated network layer. This is typically the 8B UDP header length plus the computed data length in octets, not counting the extra prefix or padding suffix octets of the virtual UDP TPDU.

If the computed UDP checksum is zero, it is represented in the UDP TPDU as a one's-complement -0 (i.e., minus zero, 0xFFFF or all ones). When conveyed by IPv4, a transmitted checksum value of a one's-complement +0 (i.e., plus zero, 0x0000 or all zeros) means that the TPDU originator did not compute a checksum (e.g., to simplify debugging, or for application protocols that can tolerate errors in the UDP data); therefore when conveyed by IPv4 a received checksum of +0 SHOULD cause the receiver to not compute and check the checksum.

A checksum value of +0 is never permitted when IPv6 conveys UDP, so for IPv6 a received checksum value of +0 indicates an erroneous UDP TPDU.

4.2.4 Optional TPDU components and elements of procedure

There are no optional TPDU components.

The behavior on receipt of a UDP TPDU conveyed by IPv4 that contains a checksum with the value +0 is unspecified by RFC768. Presumably such a UDP TPDU should be accepted, but reported with a status that conveys the fact that the TPDU was not protected during transit by any transport-layer integrity check.

4.3 Mandatory and optional protocol features

The mandatory features of the UDP protocol are

M1) The TPDU SHALL include the entire UDP header (e.g., a minimum of eight octets).

M2) The TPDU's Destination port field SHALL specify a port associated with the intended application protocol, and not a reserved port per [Port_numbers] .

M3) The TPDU header's Length field SHALL specify the full payload size of the conveying unfragmented NPDU (i.e., after any necessary NPDU reassembly) as an unsigned value.

M4) A computed checksum that has the value zero SHALL be represented in the UDP header as -0 (minus zero, 0xFFFF).

M5) TPDUs received with a checksum value other than +0 (plus zero, 0x0000), where that checksum value differs from a checksum computed equivalently across the TPDU (after virtually zeroing the TPDU's checksum field), SHALL be discarded, per RFC1122, 4.1.3.4.

M6) When conveyed by IPv6, TPDUs received with a checksum value of +0 (plus zero, 0x0000), SHALL be discarded, per RFC2460, 8.1.

M7) UDP SHALL pass IP-layer options transparently between the network and application layers, per RFC1122, 4.1.3.2.

M8) UDP SHALL pass to its associated application user all ICMP error messages it receives from the network layer, per RFC1122, 4.1.3.3.

M9) A received UDP TPDU with an IP multicast or broadcast source address SHALL be discarded without notification, per RFC1122, 4.1.3.6.

The optional (i.e., conditionally present) features of the UDP protocol are

C1) When conveyed by IPv4, the value of the TPDU header's Checksum field

i) SHOULD be other than +0 (0x0000), computed as specified in 4.2.3.2; but

ii) MAY be +0 (0x0000), indicating that checksum protection is not provided.

C2) A UDP TPDU addressed to a closed or reserved port SHOULD be replied to with an ICMP Port Unreachable PDU indicating the error, per RFC1122, 4.1.3.1.

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 error PDU in reply.

NOTE A firewall internal to the DUT, or interposed between the TD and DUT, which employs an outbound filtering ruleset also may cause discard of such ICMP error PDUs.

5 Elements of other protocols required for the testing

5.1 Protocol(s) from inferior layers used by this protocol

The UDP protocol of RFC768 is specified to operate over IPv4, because the virtual UDP TPDUs of Figure 2 use information from the conveying IPv4 NPDU. The destination-unreachable and parameter-problem

ICMP error PDUs of RFC792 (as amended) are used to report error conditions in received UDP TPDUs that are detected by the DUT's UDP implementation.

NOTE For UDP robustness testing, there is no requirement for the DUT to be able to receive ICMP PDUs, or to transmit ICMP PDUs of types other than the two just enumerated.

RFC2460, 8.1 specifies how UDP is adapted to work over IPv6, by using the replacement virtual UDP TPDU of Figure 3 that uses information from the conveying IPv6 NPDU, and by prohibiting the use of plus zero (+0000) TPDU checksums. For IPv6, RFC4443 (as amended) is the corresponding controlling ICMP specification.

5.2 Protocol(s) from superior layers used to test this protocol

Testing of UDP robustness requires only transmission of UDP test TPDUs to the DUT. Thus there is no requirement for a superior layer protocol during UDP robustness testing.

6 Robustness testing

6.1 Goals that drive testing requirements

The goal of the tests described in this document is to assess:

a) the robustness of an embedded control device with an implemented set of protocols, and

b) the device's resistance to attack, including the impact on the device's reporting and control functions while sustaining such an attack.

It is not a goal to determine the correctness of the implementation of those protocols, which would be a measure of their conformance to the requirements of the various protocol specifications.

This atypical testing goal interacts with vendor decisions to provide only partial implementations of protocols that are used within a proprietary or constrained context, such that those implementations are completely functional within the usage limits imposed by that context but are not conformant to the mandatory requirements of the controlling protocol standard.

As described by specific requirements in [EDSA-310], the consequent requirement is for this testing to

1) ascertain whether the DUT and other parts of the test configuration meet normal operational expectations before testing commences;

2) determine whether the DUT can survive receipt of invalid frames while continuing to function as expected in an automation environment; and

3) determine whether the DUT can sustain intervals of high and excessive communications load.

6.2 Testing overview

The DUT must be preconditioned to support testing by meeting the requirements of [EDSA-310] for demonstrating continued correct operation during testing.

Robustness testing occurs in three conceptual phases that may overlap, plus a test environment preconditioning phase.

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 for its ability to not evidence harm due to receipt of arbitrary erroneous frames, either singly or in combination.

   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 6 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 incorporating PDUs with varying characteristics, there is no requirement that an actual robustness test process work in this ordered, sequential manner; any order of testing is permitted provided that the selected order does not lead to incorrect conclusions about robustness.

Requirement UDP.R1 – Criteria for robustness test failure

Pass or fail of basic robustness and load stress testing SHALL be determined by:

- whether or not essential services are adequately maintained under network traffic conditions created under these tests, as defined in [CRT.Essential_services];
- any particular conditions resulting in pass/fail mandated by the testing specified in this document.

The UDP protocol that is the subject of this specification is a stateless protocol without any query/ response mechanisms.

6.3 Protocol stack used for testing

6.3.1 Protocol(s) from inferior layers used by this protocol

IP is used to convey UDP TPDUs. The virtual UDP TPDU of Figure 2 or Figure 3 uses information from the conveying IP NPDU. Although this specification presumes that IPv4 NPDUs are being conveyed by “Ethernet” DPDUs, other means of conveying UDP TPDUs, such as IPv6 over 6LoWPAN over the ISA100.11a data-link layer, are not inherently precluded.

The initial EDSA protocols covered by CRT include IPv4 rather than IPv6 or other networking protocols. The version of UDP specified in RFC768 (as amended) and tested under this test specification is UDP over IPv4 – that selection impacts the address field values used in the calculation of the UDP checksum. This document also specifies the necessary adaptation of test implementations of UDP that operate over IPv6.

6.3.2 Protocol(s) from superior layers used to test this protocol

Testing of UDP robustness requires only transmission of UDP test TPDUs to the DUT. Thus there is no requirement for a superior layer protocol during UDP robustness testing.

6.4 Phase 0: DUT preconditioning

Requirement UDP.R2 – Preconditioning of DUT and TD

The DUT, the TD(s) and possibly other devices in the test system SHALL be configured to allow observation of the performance of *essential services* of the embedded device under the test conditions, per the requirements in [CRT.Essential_services].

Essential services as defined in [CRT.Essential_services] include control loops, commands to control device configuration such as setpoints, and process alarms. A key approach to obtain observability is to use, as part of the test configuration, other automation system elements that have been engineered to communicate with and monitor the DUT.

6.5 Phase 1: Baseline operation

6.5.1 General

Requirement UDP.R3 – Demonstration of baseline operation

Before the TD commences robustness testing, the DUT shall demonstrate its ability to operate as expected in the test environment, including that the UDP component of the DUT's protocol stack is present and functioning, and that the DUT can maintain essential services.

6.5.2 Presence of proprietary protocol extensions

It is common practice for vendors to extend a standard protocol in a proprietary manner to provide functionality not covered by the standard protocol, or to provide more efficient or more constrained data transport for specific device information (e.g., when multiple device parameters require atomic update or readout as a group to maintain their inter-parameter consistency). Such extensions may take the form of extra message types, extra fields in standard messages, or extra functionality for standard fields in standard messages.

NOTE Robustness testing is not required to include specialized testing of proprietary protocol extensions. Rather, vendor disclosure of such extensions is intended to provide a basis for explanation of otherwise anomalous test results.

Requirement UDP.R4 – Equipment vendor disclosure of proprietary protocol extensions

When a protocol offered for testing has been implemented with deliberate proprietary extensions, the vendor SHALL document the extensions in a manner similar to that of Clause 4, such that robustness testing can explore the intended and unintended consequences of those protocol extensions. It is acceptable that access to this proprietary information be covered by a non-disclosure agreement (NDA) between the equipment vendor and the organization that is providing the ISCI robustness testing service.

6.6 Phase 2: Basic robustness testing

6.6.1 General

Areas of specific robustness testing are identified by analysis of the controlling protocol standards. These include identification of all field value ranges and of the bounding values of the underlying message representation (e.g., a range of 10..100 in a one-byte field, whose underlying representational bounding values are 0..255). Basic robustness testing includes testing the acceptability of each of these bounding values, and of the acceptance or rejection of adjacent values to those bounding values when such adjacent values can be represented in the message encoding. It also includes testing whether fields specified to convey signed or unsigned values are distinguished and processed appropriately.

Conceptually, basic robustness testing consists of the following, where volume or rate of message traffic is not a factor:

a) tests of valid message traffic:
   1) in expected sequences, sent at a low rate;
   2) in unexpected but valid sequences sent at a low rate (i.e., where the messages would be considered valid for the protocol under some conditions, but are not expected for the particular protocol state, message sequence or relative time);
b) tests of low rate erroneous message traffic (e.g., the ability of the device to function after receiving erroneous messages), including:
   1) single erroneous messages, including messages with inconsistent field values;
   2) properly formed messages in erroneous sequences
   3) sequences of erroneous messages.

[EDSA-310] describes the criteria for adequate performance of device essential services under these network traffic conditions. These criteria depend upon the specific service as well as whether the service operates on the same network interface used for test traffic.

6.6.2 Basis for UDP robustness testing

Correctly and incorrectly formed UDP TPDUs sent to the DUT from the DUT form the basis for UDP robustness testing.

Requirement UDP.R5 – Testing of each message field for sensitivity to invalid content

For basic robustness testing requiring erroneous messages or message sequences, valid UDP TPDUs or TPDU sequences from the TD to the DUT SHALL be altered so that one component of the UDP TPDU is erroneous; or so that the UDP TPDU is in violation of 4.3, M1 through M6 or M9; or that it is both erroneous and in violation.

Such alterations SHALL be applied to each field of the UDP TPDU where alteration might have an impact on the DUT.

NOTE 1 This type of testing can be described as single-message protocol “fuzzing”.

NOTE 2 It is the UDP protocol itself that is being tested, not any conveyed higher-level protocol.

It is suggested that basic robustness testing proceed in stages, from simple to complex, as enumerated in 6.6.1 and indicated by the following list. In general, such ordering simplifies the task of locating the source(s) of software or hardware problems should they be uncovered by the testing. However, such ordering is not a requirement.

Requirement UDP.R6 – Constituent elements in basic robustness tests

Basic UDP robustness testing SHALL include the following elements, at low traffic rates, either in distinct test phases or intermixed in a form of the test supplier’s choosing:

a) valid message traffic

b) erroneous messages

6.7 Phase 3: Load stress testing

6.7.1 General

NOTE 1 This testing phase is used to ascertain resistance to busy plant conditions as well as deliberate attacks.

Conceptually, load stress testing consists of tests of valid message traffic sent in two distinct phases:

Phase 1 – Valid message traffic is sent at a high rate less than the saturation rate threshold specified by the DUT vendor (e.g., simulating normal but busy plant conditions).

Phase 2 – Valid message traffic is sent at up to the full auto-negotiated link rate (e.g., simulating an attack or malfunction of some kind);

Attacks against a protocol implementation take the form of repeated probing by malformed messages, or by correctly formed messages whose arrival sequence and relative timing are controlled by the attacker, or (more usually) by combinations thereof, all with the intent of exploiting some oversight or error in the specific protocol implementation(s), or of activating some intertwining aspects of a multi-layer protocol stack that were unconsidered by the implementing organization.

NOTE 2 Self-induced accidental attacks are also possible, due to designer or operator oversight.

Common examples of exploited oversights and errors are deliberate buffer overflows where the implementer had neglected to detect excessive message or field size, or recursive activation of character escape encoding when the implementer had not considered recursion. Implementation interactions within a multi-layer protocol stack may occur when an initial resource allocation (e.g., memory buffering) made by one protocol layer implementation is driven into an adjustment phase that conflicts with a resource allocation already made by a paired protocol layer implementation.

6.7.2 Basis for load stress testing

Device defenses against high traffic rates impact load stress testing, and are documented by the device vendor per the following requirement.

Requirement UDP.R7 – Documentation of self-protective rate limiting behavior

Where the DUT vendor imposes rate limiting on one or more of the protocols in the test process (e.g., “Ethernet”, IP or UDP), the DUT vendor SHALL document that rate limiting occurs for that identified protocol when message rates exceed a perhaps-unspecified rate, as required by [CRT.Rate_limiting].

NOTE 1 The “Ethernet” protocol is included in this list as an identifiable placeholder for any physical and data-link protocols used to convey IPv4 or IPv6 NPDUs.

Requirement UDP.R8 – Constituent elements in load stress tests

Load stress testing SHALL include the following elements, either in distinct test phases or intermixed in a form of the test supplier’s choosing:

a) high-rate valid message traffic;

b) over-saturation-rate version of a), at the maximum auto-negotiated link rate that the TD can support.

Requirement UDP.R9 – Testing of saturation rate-limiting mechanism(s)

Saturation rate testing SHOULD be for durations of at least tens of seconds, long enough for any saturation effects to manifest. Tests that inherently involve a large number of TPDUs, such as port scans, may need to run for much longer durations so that they do not cause other untoward impact on the test environment, which inherently involves the DUT, the TD and any other devices used in ascertaining the continuing performance of the DUT’s other normal functionality (e.g., interactions with superior or peer automation system components).

Requirement UDP.R10 – Reproducibility of robustness testing

Basic robustness testing SHALL use a deterministic selection process (e.g., an offline test case generator or a seeded pseudo-random selection process) that tests combinations of valid and erroneous messages. See Clause 7 for specific required test cases.

Load stress testing SHALL use a deterministic selection process (e.g., an offline test case generator or a seeded pseudo-random selection process) that tests series of valid messages. See Clause 7 for specific required test cases.

NOTE 2 The above constraint to use of a deterministic selection process does not prohibit use of feedback from analysis of DUT responses (and non-responses) as a means of further varying and focusing testing. Nor does it prohibit use of tester-selectable options and modes to determine the aggressiveness of the test process. Rather, it is merely an attempt to facilitate reproducibility by requiring use of reproducible means to select the order, sequence and components of each test.

6.7.3 Specific load stress testing

Due to its simplicity and statelessness, the only specific feature of the UDP protocol that requires special attention is the interaction between the UDP header’s length field and the size of the payload of the conveying post-reassembly NPDU.

6.8 Reproducibility

Requirement UDP.R11 – Overall reproducibility

Baseline operation, basic robustness testing, and load stress testing SHALL be reproducible per the requirements of [CRT.Reproducibility]

Those requirements recognize that deterministic behavior of the DUT itself is not under the control of the tester and must be assumed. Further, it is acceptable to branch a test process based upon prior results.

Thus a change to the DUT may impact repeatability of a test even if the change does not intentionally cause variance for that test.

7 Specific test cases

Requirement UDP.R12 – Specific test cases

The tested suite of protocols SHALL be documented in at least the detail specified by Table 1.

Table 1 – UDP: Protocols used in test process

Protocol layer tested	Permissible alternatives	Protocols tested	Maximum load at which deliberate limiting occurs
Physical layer	IEEE 802.3		
Data-link layer	"Ethernet"		
Network layer	IPv4 + ICMPv4 error reporting or IPv6 + ICMPv6 error reporting		
Transport layer	UDP		

Requirement UDP.R13 – Testing SHALL include at least that specified by Table 2 through Table 11

These tables are descriptive, not proscriptive – there is no requirement that conforming robustness testing actually employ test sequences that are ordered or grouped as described in these tables.

Table 2 – UDP.T00: Baseline operation

Test ID	UDP.T00
Test name	Baseline operation
Test description	The basic operational aspects of the protocol under test, and of any inferior supporting protocols used in the testing, shall be demonstrated as a means of checking that gross configuration or other errors are not interfering with the testing process, that UDP is a functioning part of the DUT's protocol stack, and that the protocol implementation under test performs approximately as expected when not under test
Reference requirements	Requirement UDP.R3
Test type	Baseline operation
Test status	Mandatory
Expected DUT behavior	The DUT demonstrates basic protocol operability in the test configuration
Test object	To validate the lack of major errors in the configuration of the DUT and test environment
Test configuration	A TD is connected to the DUT by an underlying -switched network that uses IEEE 802 and IP addressing, as specified in [CRT.Test_configuration_1]
Test procedure	The TD establishes that DUT is reachable and functions normally in the test environment, before protocol-specific testing commences
Expected DUT response	The DUT demonstrates expected behavior in its "automation" environment, including that the UDP component of the protocol stack is present and functioning and that the DUT can adequately maintain essential services
Ultimate results	Pass or fail
Remarks	Initial failure of this test indicates a probable problem with the configuration of the TD or the test environment

Table 3 – UDP.T01: Truncated TPDU header with "non-negative" length field

Test ID	UDP.T01
Test name	Truncated TPDU header with "non-negative" length field
Test description	A UDP TPDU is sent as an IP NPDU payload, where the payload is a correctly formed UDP TPDU whose length field has an erroneous value between 0 and 7, inclusive, but which is well-formed through its first eight octets, and whose checksum (in octets 7 and 8 of the containing IP NPDU's payload) is correctly calculated for the specified length
Reference requirements	Requirement UDP.R5, violating 4.3, M1 and M3
Test type	Basic robustness: PDU structural violations
Test status	Mandatory
Expected DUT behavior	The DUT checks the TPDU's self-proclaimed length before checksum validation
Test object	To probe the robustness of the DUT's parsing of UDP TPDUs and protection against malformed TPDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]. ICMP error reporting by the DUT SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends an invalid UDP TPDU such that the header's length field's value is less than 8, but the conveying NPDU IP payload is otherwise a valid UDP TPDU. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	The DUT is expected to reply with an ICMP ParameterProblem (type 0x04) PDU

Table 4 – UDP.T02: Truncated TPDU header with "negative" length field

Test ID	UDP.T02
Test name	Truncated TPDU header with "negative" length field
Test description	A UDP TPDU is sent as an IP NPDU payload, where the payload is a correctly formed UDP TPDU whose length field has an erroneous value of 0xFFFF, but which is well-formed through its first eight octets
Reference requirements	Requirement UDP.R5, violating 4.3, M1 and M3
Test type	Basic robustness: PDU structural violations
Test status	Mandatory
Expected DUT behavior	The DUT checks the TPDU's self-proclaimed length (as an unsigned value) before memory allocation
Test object	To probe the robustness of the DUT's parsing of UDP TPDUs and protection against malformed TPDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]. ICMP error reporting by the DUT SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends an invalid UDP TPDU such that the header's length field's value is less than zero (when erroneously interpreted as a 2's-complement signed number) but the conveying NPDU IP payload contains an otherwise-valid UDP TPDU header. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	The DUT is expected to interpret this as a too-large TPDU, greater than the conveying IP NPDU's payload, causing it to reply with an ICMP ParameterProblem (type 0x04) PDU

Table 5 – UDP.T03: Valid TPDU shorter than IP NPDU payload

Test ID	UDP.T03
Test name	Valid TPDU shorter than IP NPDU payload
Test description	A UDP TPDU is sent whose length field value is valid but is less than the size of the delivering IP NPDU's payload
Reference requirements	Requirement UDP.R5, violating 4.3, M3
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT uses the TPDU's self-proclaimed length rather than the size of the conveying IP NPDU's payload
Test object	To probe the robustness of the DUT's parsing of UDP TPDUs and protection against malformed TPDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]. ICMP error reporting by the DUT SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends a valid UDP TPDU such that the header's length field's value is at least 8 but less than the size of the conveying IP NPDU's payload. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	1) This test might expose whether IP NPDU payload length or the UDP TPDU's explicit length is dominant during receipt processing of well-formed-TPDUs 2) The DUT is expected to process the received UDP TPDU, ignoring the extra octets in the IP NPDU payload

Table 6 – UDP.T04: Truncated TPDU

Test ID	UDP.T04
Test name	Truncated TPDU
Test description	A UDP TPDU is sent whose length field value is greater than the size of the delivering IP NPDU's payload
Reference requirements	Requirement UDP.R5, violating 4.3, M3
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT uses the TPDU's self-proclaimed length rather than the size of the conveying IP NPDU's payload, which must at least equal that self-proclaimed length value
Test object	To probe the robustness of the DUT's parsing of UDP TPDUs and protection against malformed TPDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]. ICMP error reporting by the DUT SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends a truncated UDP TPDU such that the header's length field's value is at least 8 and greater than the size of the conveying IP NPDU's payload, but where the length value is 0x7FFF or less. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	1) This test might expose whether IP NPDU payload length or the UDP TPDU's explicit length is dominant during receipt processing of well-formed TPDUs. It also attempts to determine whether the DUT treats the TPDU length field as an unsigned or signed value 2) The DUT is expected to reply with an ICMP ParameterProblem (type 0x04) PDU

Table 7 – UDP.T05: TPDU length signedness

Test ID	UDP.T05
Test name	TPDU length signedness
Test description	A UDP TPDU is sent whose length field value is greater than 0x7FFF, either in a single IP NPDU or as multiple fragments of a fragmented IP NPDU
Reference requirements	Requirement UDP.R5, violating 4.3, M3
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory when the IP implementation supports unfragmented or post-reassembly NPDU payloads of size 32 KiB or greater
Expected DUT behavior	The DUT interprets the TPDU's self-proclaimed length as an unsigned value
Test object	To probe the robustness of the DUT's parsing of UDP TPDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]
Test procedure	The TD sends a valid UDP TPDU such that the header's length field's unsigned value is greater than 0x7FFF. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	1) This test might expose whether the DUT treats the TPDU length field as an unsigned or signed value. As such it may test a different aspect of TPDU processing than UDP.T02, specified in Table 4 2) The DUT is likely to reply with an ICMP ParameterProblem (type 0x04) PDU]

Table 8 – UDP.T06: Invalid TPDU checksum

Test ID	UDP.T06
Test name	Invalid TPDU checksum
Test description	UDP TPDUs are sent whose checksum field value differs from the computed checksum for the TPDU and which, when conveyed by IPv4, does not have the value +0
Reference requirements	Requirement UDP.R5, violating 4.3, M5
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT validates UDP TPDU checksums on receipt and compute UDP TPDU checksums prior to transmission. The value +0 SHALL NOT be used when the conveying network layer protocol is IPv6
Test object	To probe the robustness of the DUT's checksum processing for UDP TPDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]
Test procedure	The TD sends UDP TPDUs whose contained checksum field specifies a checksum value different than that which the DUT is expected to compute or accept on the received TPDU. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	1) This test exposes failures to compute and validate checksums correctly 2) The DUT is likely to reply with an ICMP ParameterProblem (type 0x04) PDU]

Table 9 – UDP.T07: Rejection of UDP TPDUs addressed to reserved destination ports

Test ID	UDP.T07
Test name	Rejection of UDP TPDUs addressed to reserved destination ports
Test description	Many UDP ports are reserved and not available for network communication
Reference requirements	Requirement UDP.R5, violating 4.3, M2
Test type	Basic robustness: PDU content semantic violations
Test status	Optional
Expected DUT behavior	The DUT does not use a reserved UDP port as a destination port
Test object	To probe the robustness of the DUT's use of reserved destination ports
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]. ICMP error reporting by the DUT SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends an otherwise valid UDP TPDU whose destination port is a reserved port according to [Port_numbers]. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services.
Results	Pass or fail
Remarks	1) This test is redundant to the prior UDP port scan of [EDSA-310], which should have observed rejection or ignoring of UDP TPDUs addressed to reserved ports 2) The DUT is likely to reply with an ICMP DestinationUnreachable (type 0x0B) PDU specifying a reason code of 0x03, port unreachable

Table 10 – UDP.T08: UDP conveyed-application robustness

Test ID	UDP.T08
Test name	UDP conveyed-application robustness
Test description	The TD generates UDP data and sends it to open UDP ports on the DUT. Any data size supported by UDP may be generated, from 0 B to 65,535 B of UDP payload. The UDP data uses a variety of data patterns known to cause problems for some UDP-conveyable protocols, without attempting to target any specific UDP-conveyable protocol.
Reference requirements	
Test type	Basic robustness: APDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT continues to function while receiving such UDP TPDUs, provided that the load thus induced is less than that claimed as supportable by the DUT vendor
Test object	To probe the robustness of the DUT's ability to receive and withstand a rate-limited burst of TPDUs addressed to its discovered open ports, similar to that induced by an attacker's follow-up to a port scan
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]
Test procedure	The TD sends valid UDP TPDUs conveying varying but focused APDU data addressed to various UDP ports of the DUT, at at a rate less than that at which the DUT's manufacturer claims DUT protective measures will be invoked
Expected DUT response	The DUT is expected to continue network communication even under focused load while adequately maintaining essential services
Results	Pass or fail
Remarks	

Table 11 – UDP.T09: Maintenance of service under high load, including network saturation: Raw TPDU flood

Test ID	UDP.T09
Test name	Maintenance of service under high load, including network saturation: Raw TPDU flood
Test description	A flurry of UDP TPDUs is sent to the DUT to attempt to overwhelm the DUT's receive processing and storage resources. This test proceeds in two phases: • Phase 1: as a high load test during which the DUT SHOULD respond normally to received messages • Phase 2: as a network saturation test during which the DUT MAY invoke protective behaviors such as blocking network reception but SHOULD otherwise function normally. See [CRT.Rate_limiting] for additional requirements
Reference requirements	Requirement UDP.R8
Test type	Load stress
Test status	Mandatory
Expected DUT behavior	The DUT protects itself against a flood of received UDP TPDUs. • Phase 1: The DUT continues to function, adequately maintaining all essential services, in the presence of a sudden burst of received UDP TPDUs, provided that the load thus induced is less than that claimed as supportable by the DUT vendor; • Phase 2: The DUT adequately maintains essential control, even if it must reduce or cease other essential services during the period of network overload.
Test object	To evaluate the DUT's ability to receive and withstand a burst of TPDUs addressed to it
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]. The DUT vendor SHALL state a rate limit below which protective measures are not expected to be invoked
Test procedure	The TD sends valid TPDUs that are either explicitly or implicitly addressed to the DUT • Phase 1: at a rate less than that at which the DUT's manufacturer claims DUT protective measures will be invoked; • Phase 2: at a rate up to the auto-negotiated maximum rate of the underlying network, maintains that high load rate for a few seconds, then gradually reduces its sending rate to zero. TPDUs sent to the DUT MAY be conveyed by IP packets using any of the classes of explicit or implicit IP addressing (i.e., for IPv4, unicast, broadcast and,multicast; for IPv6, unicast/anycast and multicast), in any combination. Testing SHALL use destination IP addresses that the DUT is configured to recognize, including at least one of each class of recognized IP address
Expected DUT response	• Phase 1: The DUT is expected to continue network communication even under high load while adequately maintaining essential services. • Phase 2: The DUT is expected to activate protective measures at some (vendor unspecified) level of resource demand, and to recover some reasonable time interval after that demand for resources is reduced substantially below the level at which the protective measures were triggered. The DUT is expected to adequately maintain essential control throughout the test
Results	Pass or fail
Remarks	The DUT vendor is not required to be able to predict the messaging rate at which such protective measures are invoked, but SHOULD be able to put an upper bound on time after the stimulus ceases before the recovery is complete

Bibliography

IANA protocol and number registries, http://www.iana.org/protocols/
registries of various assigned code points for standard Internet protocols

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

RFC1240, *OSI connectionless transport services on top of UDP (v.1)*
provides guidance on encoding OSI TSAPs as a subheader within a UDP payload

— — — — — —