UNH IOL iSCSI CONSORTIUM

Login Phase Test Suite for iSCSI Initiators
Version 3.0

Technical Document

Last Updated November 19, 2015

UNH-IOL iSCSI Consortium
InterOperability Laboratory
University of New Hampshire

21 Madbury Road, Suite 100
Durham, NH 03824
Phone: (603) 862-1908
Fax: (603) 862-4181
https://www.iol.unh.edu/testing/storage/iscsi

TABLE OF CONTENTS

MODIFICATION RECORD

[1]July 28, 2003 (Version 0.1) DRAFT RELEASE
David Woolf: Initial draft release to draft 20 of the iSCSI standard
[2]February 23, 2005 (Version 1.0) FINAL RELEASE
David Woolf: Test Suite updated to match final RFC 3720 standard.
Updated references and simplified discussions.
[3]April 11, 2006 (Version 1.1) FINAL RELEASE
David Woolf: Corrected test 26.3.2.
[4]January 4, 2007 (Version 1.2) FINAL RELEASE
Aaron Bascom: Changed title page.
[5]April 16, 2007 (Version 1.3) FINAL RELEASE
Aaron Bascom: Clarified and updated 7.2, 7.5, 9.2, 12.1, 13.2, 20.1, 26.1, 26.2.2, 26.3.1, 26.3.2, and 26.4.
[6]July 28, 2007 (Version 2.0) FINAL RELEASE
Ethan Burns: Test Suite updated for iSCSI Corrections and Clarifications:
Updated: 27.1.
Added 28.1, 29.1 and 30.1.
[7]August 20, 2007 (Version 2.0) FINAL RELEASE
Aaron Bascom: Added iSCSI Corrections and Clarifications to References and introduction.
[8]March 23, 2008 (Version 2.0) FINAL RELEASE
Ethan Burns: Updated acknowledgements.
Updated iSCSI Corrections and Clarifications reference to RFC 5048
[9]May 22, 2009 (Version 2.1) FINAL RELEASE
Patrick MacArthur: Split Test #24.1 into #24.1 and #24.2.
Added Test #31.1
[10] September 8, 2015 (Version 2.2) FINAL RELEASE
Andrew Johnson: Updated test suite with new manager's information.
Updated test names for uniqueness
[11] November 3, 2015 (Version 3.0) FINAL RELEASE
Amy Davies: Updated references to RFC 7143
Fix broken headings
Removed Digital Signature Information.
Updated wording of Possible Problems to new "Not Supported" wording
Updated address
Added RFC 7144 to References
Modified test #12.1 for deprecated keys
Modified test #30.1 to remove Possible Problem for unsupported TaskReporting.
Added informative test #31.1 for iSCSIProtocolLevel key
Added test #33.1 for deprecated X#, Y#, and Z# prefixes.

ACKNOWLEDGMENTS

The University of New Hampshire would like to acknowledge the efforts of the following individuals in the development of this test suite.

Dr. Bob Russell	UNH Department of Computer Science
Peter Scruton	UNH InterOperability Laboratory
David Woolf	UNH InterOperability Laboratory
Aaron Bascom	UNH InterOperability Laboratory
Ethan Burns	UNH InterOperability Laboratory
Patrick MacArthur	UNH InterOperability Laboratory
Amy Davies	UNH InterOperability Laboratory

INTRODUCTION

The University of New Hampshire's InterOperability Laboratory (IOL) is an institution designed to improve the interoperability of standards based products by providing an environment where a product can be tested against other implementations of a standard. This particular suite of tests has been developed to help implementers evaluate the Login Phase functionality of their iSCSI initiators.

These tests are designed to determine if an iSCSI product conforms to specifications defined in ***IETF RFC 7143 Internet Small Computer System Interface (iSCSI) Protocol (Consolidated)*** (hereafter referred to as the "iSCSI Standard"). Successful completion of all tests contained in this suite does not guarantee that the tested device will successfully operate with other iSCSI products. However, when combined with satisfactory operation in the IOL's interoperability test bed, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many iSCSI environments.

The tests contained in this document are organized in order to simplify the identification of information related to a test, and to facilitate in the actual testing process. Tests are separated into groups, primarily in order to reduce setup time in the lab environment, however the different groups typically also tend to focus on specific aspects of device functionality. A dot-notated naming system is used to catalog the tests, where the first number always indicates a specific group of tests in the test suite is based. The second and third numbers indicate the test's group number and test number within that group, respectively. This format allows for the addition of future tests in the appropriate groups without requiring the renumbering of the subsequent tests.

The test definitions themselves are intended to provide a high-level description of the motivation, resources, procedures, and methodologies specific to each test. Formally, each test description contains the following sections:

Purpose

The purpose is a brief statement outlining what the test attempts to achieve. The test is written at the functional level.

References

This section specifies all reference material *external* to the test suite, including the specific sub clauses references for the test in question, and any other references that might be helpful in understanding the test methodology and/or test results. External sources are always referenced by a bracketed name (e.g., [RFC-7143]) when mentioned in the test description. Any other references in the test description that are not indicated in this manner refer to elements within the test suite document itself (e.g., "Appendix 5.A", or "Table 5.1.1-1")

Resource Requirements

The requirements section specifies the test hardware and/or software needed to perform the test. This is generally expressed in terms of minimum requirements, however in some cases specific equipment manufacturer/model information may be provided.

Last Modification

This specifies the date of the last modification to this test.

Discussion

The discussion covers the assumptions made in the design or implementation of the test, as well as known limitations. Other items specific to the test are covered here as well.

Test Setup

The setup section describes the initial configuration of the test environment. Small changes in the configuration should not be included here, and are generally covered in the test procedure section (next).

Procedure

The procedure section of the test description contains the systematic instructions for carrying out the test. It provides a cookbook approach to testing, and may be interspersed with observable results.

Observable Results

This section lists the specific observables that can be examined by the tester in order to verify that the DUT is operating properly. When multiple values for an observable are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail outcome for a particular test is generally based on the successful (or unsuccessful) detection of a specific observable.

Possible Problems

This section contains a description of known issues with the test procedure, which may affect test results in certain situations. It may also refer the reader to test suite appendices and/or other external sources that may provide more detail regarding these issues.

REFERENCES

The following documents are referenced in this text:

[RFC-7143] Chadalapaka, M. Satran, J. Black, D. Internet Small Computer System Interface (iSCSI) Protocol (Consolidated). RFC 7143, April 2014

[RFC-7144] Knight, F. Chadalapaka, M. Internet Small Computer System Interface (iSCSI) SCSI Features Update. RFC 7144, April 2014

ADDITIONAL ACRONYMS AND ABBREVIATIONS

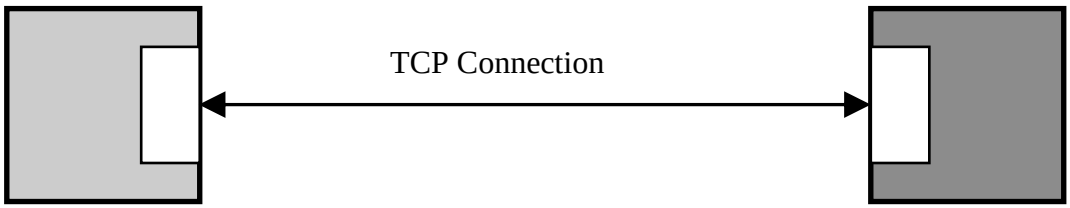
The acronyms and abbreviations defined here supplement the acronyms defined in IETF RFC 7143 section 2.1 and may be used in this document.

Acronym	Definition
DUT	Device Under Test
DSL	DataSegmentLength
MRDSL	MaxRecvDataSegmentLength

TEST SETUP

The following test setup is used in this test suite:

Test Setup 1:



Testing Station/ Monitor DUT

GROUP 1: LOGIN PHASE FOR INITIATORS

Overview: This group of tests verifies the Login Phase specifications of iSCSI defined in RFC 7143. Comments and questions regarding the implementation of these tests are welcome, and may be forwarded to Kerry Munson at the UNH InterOperability Lab (kerry.munson@iol.unh.edu).

Test #1.1 Standard Login

Purpose: To verify that the DUT properly uses the InitiatorTaskTag, CID, VersionMax, VersionMin, CmdSN, and ISID fields, in the Login Request PDU.

Reference: [RFC-7143] Section 4.6.3.2, 11.12.7, 11.12.4, 11.12.8, 11.12.5

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 4.6.3.2

> "Login Requests and Responses are used exclusively during the Login Phase of each connection to set up the session and connection parameters. (The Login Phase consists of a sequence of login requests and responses carrying the same Initiator Task Tag.)
> A connection is identified by an arbitrarily selected connection-ID (CID) that is unique within a session."

[RFC-7143] Section 11.12.7

> "The CID is a unique ID for a connection within the session.
>
> All Login requests and responses within the Login phase MUST carry the same CID."

[RFC-7143] Section 11.12.4

> "The version number for this document is 0x00. Therefore, both Version-min and Version-max MUST be set to 0x00."

[RFC-7143] Section 11.12.4.1

> "All Login Requests within the Login Phase MUST carry the same Version-max"

[RFC-7143] Section 11.12.4.2

> "All Login Requests within the Login Phase MUST carry the same Version-min."

[RFC-7143] Section 11.12.8

> "The CmdSN is either the initial command sequence number of a session (for the first Login Request of a session -- the "leading" login) or the command sequence number in the command stream if the login is for a new connection in an existing session. . . . If the leading login carries the CmdSN 123, all other Login Requests in the same Login Phase carry the CmdSN 123, and the first non-immediate command in the Full Feature Phase also carries the CmdSN 123. . . . If the Login Request is a leading Login Request, the target MUST use the value presented in the CmdSN as the target value for the ExpCmdSN."

[RFC-7143] Section 11.12.5

> The ISID is the initiator assigned portion of the SSID. The allowable formats are as follows: For T=00b, A&B are a 22 bit OUI, C&D are a 24 bit qualifier. For T=01b, A is reserved, B&C are an IANI Enterprise Number, D is a qualifier. For T=10b, A is reserved, B&C are random, D is a qualifier. For T=11b, A, B, C, D are all reserved.

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and transmit a Login Request PDU.

Observable Results:

- Verify that the Initiator offers an InitiatorTaskTag in the first Login Request PDU and that the Initiator does not change the InitiatorTaskTag in the course of the Login Phase.
- Verify that an initiator in the Login phase uses only one CID. Perform parameter negotiation in order to see multiple login PDUs.
- Verify that in the login led by the DUT, the Version Max and Version Min fields were constant in all Login Requests. The DUT must use the version number for the current version of the iSCSI RFC. The Version number for the current draft is 0x00 for both Version Max and Version Min.
- Verify that the device provides a value for CmdSN and does not increment it while in the login phase.
- Verify that the ISID field is formatted correctly.
- Verify that throughout the login phase the DUT does not use '?' as a value, indicating an inquiry. This is an invalid value.

Possible Problems: None

Test #2.1 TSIH

Purpose: To verify that the DUT properly uses the TSIH field.

Reference: [RFC-7143] Section 11.12.6

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 11.12.6

> "The TSIH must be set in the first Login Request. The reserved value 0 MUST be used on the first connection for a new session. Otherwise, the TSIH sent by the target at the conclusion of the successful login of the first connection for this session MUST be used. The TSIH identifies to the target the associated existing session for this new connection.
>
> All Login Requests within a Login Phase MUST carry the same TSIH."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and send a Login Request PDU.
- The Testing Station should transmit Login Responses to the DUT and complete the Login Phase. Each Login Response PDU the Testing Station sends will have a TSIH of 0.
- The Testing Station will send a final Login Response with the TSIH field set to a non-zero value.

Observable Results:

- Verify that an initial Login request has a TSIH of zero.
- Verify that in subsequent Login Requests, the DUT uses the TSIH field of zero.

Possible Problems: None

Test #2.2 TSIH Invalid Value

Purpose: To verify that the DUT properly checks the TSIH field.

Reference: [RFC-7143] Section 11.12.6, 11.13.3

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 11.12.6

> "The TSIH must be set in the first Login Request. The reserved value 0 MUST be used on the first connection for a new session. Otherwise, the TSIH sent by the target at the conclusion of the successful login of the first connection for this session MUST be used. The TSIH identifies to the target the associated existing session for this new connection.
>
> All Login Requests within a Login Phase MUST carry the same TSIH."

[RFC-7143] Section 11.13.3

> "With the exception of the Login Final-Response in a new session, this field should be set to the TSIH provided by the initiator in the Login Request. For a new session, the target MUST generate a non-zero TSIH and ONLY return it in the Login Final-Response"

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station.
- The Testing Station sends a Login Response with the TSIH field set to 0x7777.

Observable Results:

- Verify that an initial Login request has a TSIH of zero.
- Verify that the DUT does not use the TSIH supplied by the Testing Station, but rather continues to use a value of zero. The DUT may ignore the TSIH field or close the connection.

Possible Problems: None

Test #2.3 TSIH in Second Connection

Purpose: To verify that the DUT properly checks the TSIH field when specifying a second connection.

Reference: [RFC-7143] Section 11.12.6

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 11.12.6

> "The TSIH must be set in the first Login Request. The reserved value 0 MUST be used on the first connection for a new session. Otherwise, the TSIH sent by the target at the conclusion of the successful login of the first connection for this session MUST be used."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station
- Verify that the first connection in the session uses a TSIH of zero. The Testing Station should provide a non-zero TSIH in the Final Login Response.
- If the device supports multiple connections, allow it to add a second connection to its current session with the Testing Station.

Observable Results:

- Verify that the DUT uses the TSIH provided by the Testing Station in the Final Login Response of the first connection, in the initial Login Request of the second connection.

Possible Problems: If the DUT does not support multiple connections per session, the result of this test is "Not Supported".

Test #3.1 T-Bit

Purpose: To verify that the DUT properly uses the T-Bit field.

Reference: [RFC-7143] Section 11.12.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 11.12.1

T (Transit) Bit – "When set to 1, this bit indicates that the initiator is ready to transit to the next stage.

If the T bit is set to 1 and NSG is Full Feature Phase, then this indicates that the initiator is ready for the Final Login Response"

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station. Wait for the initiator to transmit a login request to the Testing Station with the T Bit set to 1. Verify that NSG is set to a value higher than CSG in this login request.
- The Testing Station should respond with a login response with the T Bit set to zero. If possible the Testing Station should include parameters to be negotiated in this Login Response. Wait for the initiator to transmit another login request.
- If this second received Login Request has the T Bit set to 1, the Testing Station should transmit a login response with the T Bit set to 1, and move on to the Full Feature Phase.

Observable Results:

- Verify that when T=1 and NSG=Full Feature Phase from the DUT, the Testing Station transmits a Login Final Response, and the DUT moves on to Full Feature Phase Verify that the DUT moves into Full Feature Phase operation by looking for SCSI Commands.
- When the Testing Station delays moving into Full Feature Phase by not sending a Final Login Response when the DUT has set T=1 and NSG=Full Feature Phase, verify that the initiator does not change its value of CSG from the previous login request when it responds to the received Login Response.

Possible Problems: None

Test #3.2 T-Bit Delay

Purpose: To verify that the DUT properly uses the T-Bit field.

Reference: [RFC-7143] Section 11.12.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 11.12.1

T (Transit) Bit – "If the T bit is set to 1 and NSG is Full Feature Phase, then this indicates that the initiator is ready for the Final Login Response"

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and transmit an Initial Login Request. The Testing Station should transmit Login Responses to complete the Login Phase.
- Wait for the initiator to transmit a login request to the Testing Station with the T Bit set to 1. The Testing Station should respond with a login response with the T Bit set to zero and no parameters included for negotiation. Wait for the initiator to transmit another login request.
- The Testing Station should transmit a login response with the T Bit set to 0 to any subsequent login requests from the DUT, whether the T bit is set or not, repeat 5 times.
- The Testing Station should wait for the initiator to transmit a login request with the T Bit set to 1 and reply with a Login Response with the T bit set to 1.

Observable Results:

- Verify that when the DUT transmits a Login Request PDU with T=1, that NSG is set to a value higher than CSG.
- Verify that in Login Requests after the DUT has set T=1, the DUT does not change its value of CSG from the previous login requests.
- Verify that when the T=1 and NSG=Full Feature Phase, and the next Login Response from the Testing Station with T=1 is the final Login Response, and both devices move into the Full Feature Phase. This can be seen by if the DUT begins sourcing SCSI commands.

Possible Problems: None

Test #4.1 StatSN Zero

Purpose: To verify that the DUT properly uses the StatSN field.

Reference: [RFC-7143] Section 11.13.4

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 11.13.4

> "For the first Login Response (the response to the first Login Request), this is the starting status sequence number for the connection. The next response of any kind -- including the next Login Response, if any, in the same Login Phase -- will carry this number + 1. This field is only valid if the Status-Class is 0."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station.
- The Testing Station should transmit a login response with StatSN set to 0.

Observable Results: Verify that the DUT initializes ExpStatSN to 1, which is one greater than the StatSN value provided in the First Login Response from the Testing Station.

Possible Problems: None

Test #4.2 StatSN Non-Zero

Purpose: To verify that the DUT properly uses the StatSN field.

Reference: [RFC-7143] Section 11.13.4

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 11.13.4

> "For the first Login Response (the response to the first Login Request), this is the starting status sequence number for the connection. The next response of any kind -- including the next Login Response, if any, in the same Login Phase -- will carry this number + 1. This field is only valid if the Status-Class is 0."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station.
- The Testing Station should transmit a login request with StatSN set to 123.

Observable Results:
- Verify that the DUT initializes ExpStatSN to 124, which is one greater than the value provided in the Login Response from the Testing Station.

Possible Problems: None

Test #5.1 ExpStatSN

Purpose: To verify that the DUT properly uses the ExpStatSN field.

Reference: [RFC-7143] Section 11.12.9

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 11.12.9

> "For the first Login Request on a connection this is ExpStatSN for the old connection and this field is only valid if the Login request restarts a connection
>
> For subsequent Login Requests it is used to acknowledge the Login Responses with their increasing StatSN values."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and transmit a Login Request PDU.

Observable Results:

- Verify that in the first Login Request transmitted by the DUT it leaves ExpStatSN at zero, since connection reinstatement is not occurring.

Possible Problems: The DUT may not attempt a connection restart, or may only do so if the connection is terminated while in the Full Feature Phase.

Test #6.1 Negotiate Once

Purpose: To verify that the DUT allows key=value pairs to only be negotiated once in any given login phase, and that key=value pairs are properly followed by one null character.

Reference: [RFC-7143] Section 6.3

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.3

> "Neither the initiator nor the target should attempt to declare or negotiate a parameter more than once during login except for responses to specific keys that explicitly allow repeated key declarations (e.g., TargetAddress). An attempt to renegotiate/redeclare parameters not specifically allowed MUST be detected by the initiator and target. . . . if detected by the initiator, the initiator MUST drop the connection."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and transmit a First Login Request.
- The Testing Station should transmit a Login Response with appropriate responses to each of the key=value pairs offered by the DUT in the First Login Request.

Observable Results:

- Verify that once a particular parameter negotiation is complete, that it is not offered again during the login.
- Verify that all key=value pairs offered, are followed by one null (0x00) character.

Possible Problems: None.

Test #6.2 Negotiate Once ImmediateData Separate PDUs

Purpose: To verify that the DUT allows key=value pairs to only be negotiated once in any given login phase.

Reference: [RFC-7143] Section 6.3

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.3

> "Neither the initiator nor the target should attempt to declare or negotiate a parameter more than once during login except for responses to specific keys that explicitly allow repeated key declarations (e.g., TargetAddress). An attempt to renegotiate/redeclare parameters not specifically allowed MUST be detected by the initiator and target. . . . if detected by the initiator, the initiator MUST drop the connection."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and performed a Standard Login.
- The Testing Station should offer the Immediate Data parameter twice during the Operational Parameter Negotiation in separate PDUs.

Observable Results:

- Verify that the device terminates the connection on seeing the ImmediateData key twice.

Possible Problems: Some devices may choose to not 'detect' the occurrence of a renegotiation, viewing detection of such an error as optional. This is not the intent of the text at section 6.3. If a initiator chooses to not drop a connection where it is receiving parameters for renegotiation, its leaves itself open for a denial of service attack

Test #6.3 Negotiate Once ImmediateData Same PDU

Purpose: To verify that the DUT allows key=value pairs to only be negotiated once in any given login phase.

Reference: [RFC-7143] Section 6.3

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.3

> "Neither the initiator nor the target should attempt to declare or negotiate a parameter more than once during login except for responses to specific keys that explicitly allow repeated key declarations (e.g., TargetAddress). An attempt to renegotiate/redeclare parameters not specifically allowed MUST be detected by the initiator and target. . . . if detected by the initiator, the initiator MUST drop the connection."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and performed a Standard Login.
- The Testing Station should offer the Immediate Data parameter twice during the Operational Parameter Negotiation in the same PDU.

Observable Results:

- Verify that the device terminates the connection on seeing the ImmediateData key twice.

Possible Problems: Some devices may choose to not 'detect' the occurrence of a renegotiation, viewing detection of such an error as optional. This is not the intent of the text at section 6.3. If a target chooses to not drop a connection where it is receiving parameters for renegotiation, its leaves itself open for a denial of service attack

Test #7.1 Login Request

Purpose: To verify that the DUT includes the proper information in the Initial Request of the Login phase.

Reference: [RFC-7143] Section 6.3

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.3

> "The initial Login Request of any connection MUST include the InitiatorName key=value pair. . . . For any connection within a session whose type is not "Discovery", the first Login Request MUST also include the TargetName key=value pair. . . . The Login Phase MAY include a SecurityNegotiation stage and a LoginOperationalNegotiation stage and MUST include at least one of them, but the included stage MAY be empty except for the mandatory names. . . . If both stages are used, the SecurityNegotiation MUST precede the LoginOperationalNegotiation."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login. This connection must not be within a Discovery Session.

Observable Results:

- Verify that the DUT performs an Initial Login Request.
- Verify that the DUT does not move directly into the Full Feature Phase.
- Verify that the DUT does not attempt to go to Security stage after entering Operational Negotiation stage.
- Verify that the Initial Login Request includes the InitiatorName and TargetName keys, the protocol version supported, session and connection ID, and the negotiation stage that the initiator is ready to enter, if the DUT has set the T=1.

Possible Problems: None.

Test #7.2: Login Request List Negotiation

Purpose: To verify that the DUT handles the negotiation in the Login Phase correctly.

Reference: [RFC-7143] Section 6.2.1, 13.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6.2.1

> "In list negotiation, the originator sends a list of values (which may include "None") in its order of preference.
>
> The responding party MUST respond with the same key and the first value that it supports (and is allowed to use for the specific originator) selected from the originator list."

[RFC-7143] Section 13.1

> HeaderDigest and DataDigest – "checksums that can be negotiated for the digests and MUST be implemented by every iSCSI initiator and target": CRC32C, None

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT. Wait for the DUT to begin the Login Phase and Operational Parameter negotiation.
- The Testing Station should transmit a Login Response PDU with any keys offered by the DUT and if not already included, also one of the following key=list pairs: HeaderDigest or DataDigest=Y-1.unh.edu, Y-2.unh.edu, Y-3.unh.edu, CRC32C, None.

Observable Results:

- Verify that the DUT responds with the first value it supports and ignores all other values.

Possible Problems: If the DUT chooses to offer both HeaderDigest and DataDigest in the first Login Request, the result of this test is "Not Supported".

Test #7.3: Login Request Invalid AuthMethod

Purpose: To verify that the DUT handles the negotiation in the Login Phase correctly.

Reference: [RFC-7143] Section 6.2.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.2.1

> "In list negotiation, the originator sends a list of values (which may include "None") in its order of preference.
>
> The responding party MUST respond with the same key and the first value that it supports (and is allowed to use for the specific originator) selected from the originator list. . . . The selection of a value not proposed MUST be handled by the originator as a protocol error."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT.
- Wait for the DUT to begin the Login Phase and Security Negotiation. The DUT is expected to offer some key=value pair for AuthMethod.
- The Testing Station should transmit a Login Response PDU with the AuthMethod key and a value not offered by the DUT.

Observable Results:

- Verify that the DUT as the requester recognizes this as a failed negotiation and proceeds by dropping the connection.

Possible Problems: The DUT may 'skip' Security Negotiation and proceed directly to Operational Parameter negotiation.

Test #7.4: Login Request FirstBurstLength Value Too High

Purpose: To verify that the DUT handles the negotiation in the Login Phase correctly.

Reference: [RFC-7143] Section 6.2.2, 13.14

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.2.2

> "For simple-value negotiations, the accepting party MUST answer with the same key. The value it selects becomes the negotiation result."

[RFC-7143] Section 13.14

> FirstBurstLength is defined as a numerical value between 512 and 2^{24} - 1

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT. Wait for the DUT to begin the Login Phase and Operational Parameter negotiation.
- The Testing Station should offer values for Immediate Data and InitialR2T so that FirstBurstLength is relevant.
- The Testing Station should transmit a Login Response PDU with the following key=value pair: FirstBurstLength = 16777216, if FirstBurstLength was not already offered. This value is higher than the maximum legal value for FirstBurstLength.

Observable Results:

- Verify that the device responds with a value of Reject and/or terminates the connection, or replies with a number within the valid range for FirstBurstLength.

Possible Problems: If FirstBurstLength is Irrelevant, the result of this test is "Not Supported".

Test #7.5: Login Request ImmediateData Invalid Value

Purpose: To verify that the DUT handles the negotiation in the Login Phase correctly.

Reference: [RFC-7143] Section 6.2, 6.2.2

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.2

> "The general format of text negotiation is:
> Proposer-> <key>=<valuex>
> Acceptor-> <key>={<valuey> | NotUnderstood | Irrelevant | Reject}"

[RFC-7143] Section 6.2.2

> "For Boolean negotiations (i.e., keys taking the values "Yes" or "No"), the accepting party MUST answer with the same key and the result of the negotiation"

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT. Wait for the DUT to begin the Login Phase and Operational Parameter negotiation.
- The Testing Station should transmit a Login Response PDU with the following key=value pair: Immediate Data=Ok.

Observable Results:

- If ImmediateData was not offered by the DUT, then verify that the device responds with the key-value pair Immediate Data=Reject or selects an admissible value.
- The DUT may also disconnect due to a protocol error.

Possible Problems: None.

Test #7.6: Login Request Invalid Key

Purpose: To verify that the DUT handles the negotiation in the Login Phase correctly.

Reference: [RFC-7143] Section 6.2

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.2

> "In Login Phase negotiation, any key not understood by the acceptor may be ignored by the acceptor without affecting the basic function. However, the answer for a key not understood MUST be key=NotUnderstood."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT.
- The Testing Station should transmit a Login Request PDU with the following key=value pair: ImmediateDate=Yes. Note that the key is invalid.

Observable Results:

- Verify that the device responds with the key=value pair ImmediateDate=NotUnderstood.

Possible Problems: None.

Test #8.1 Invalid PDU

Purpose: To verify that the DUT properly identifies and reacts to an Invalid PDU.

Reference: [RFC-7143] Section 4.2.4

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 4.2.4

> "Only login request and response PDUs are allowed in the login phase. If an initiator receives any PDU except a Login response, it MUST immediately terminate the connection."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit a Data-In PDU to the device.

Observable Results:

- Verify that the device terminates the connection.

Possible Problems: None.

Test #9.1 Header and Data Digests Offer

Purpose: To verify that the DUT properly negotiates Header and Data Digests.

Reference: [RFC-7143] Section 13.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.1
HeaderDigest and DataDigest – "checksums that can be negotiated for the digests and MUST be implemented by every iSCSI initiator and target": CRC32C, None

[RFC-7143] Section 13.1
"Private or public extension algorithms MAY also be negotiated for digests."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station and perform a Standard Login.

Observable Results:
- Verify if the DUT attempts a Header or Data Digest negotiation, it offers 'CRC32C' and/or 'None' as options.

Possible Problems: None.

Test #9.2: Header and Data Digest Response

Purpose: To verify that the DUT properly negotiates values for Header and Data Digests. Even if the response will be 'None' the DUT must transmit a response.

Reference: [RFC-7143] Section 6.2.1, 13.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6.2.1

> "In list negotiation, the originator sends a list of values (which may include "None"), in order of preference.
>
> The responding party MUST respond with the same key and the first value that it supports (and is allowed to use for the specific originator) selected from the originator list."

[RFC-7143] Section 13.1

> HeaderDigest and DataDigest – "checksums that can be negotiated for the digests and MUST be implemented by every iSCSI initiator and target": CRC32C, None

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the DUT to connect to the Testing Station.
- Perform a Standard Login
- If not already offered, offer the following key=value pair: HeaderDigest=AwesomeAlgorithm, None.

Observable Results:

- Verify that the device responds with the value 'None'.

Possible Problems: If the DUT offers HeaderDigest in the leading login, the result of this test is "Not Supported".

Test #9.3 Header and Data Digests Support CRC32C

Purpose: To verify that the DUT supports CRC32C for Header and Data Digest.

Reference: [RFC-7143] Section 13.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.1

HeaderDigest and DataDigest – "checksums that can be negotiated for the digests and MUST be implemented by every iSCSI initiator and target": CRC32C, None

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit the Key=Value pairs: HeaderDigest=CRC32C, DataDigest=CRC32C

Observable Results:

- Verify that if the DUT attempts a Header or Data Digest negotiation, it offers 'CRC32C' as a possible option.
- Verify that the DUT does not reject the key=value pairs HeaderDigest=CRC32C or DataDigest=CRC32C.
- Verify that the DUT does not close the connection.

Possible Problems: None.

Test #10.1 MaxConnections

Purpose: To verify that the DUT properly negotiates MaxConnections.

Reference: [RFC-7143] Section 13.2

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13, 13.2
MaxConnections is defined as "Use: LO", meaning that it "MUST only be carried on the leading connection and cannot be changed after the leading connection login"

[RFC-7143] Section 13.2
MaxConnections is defined as a numerical value between 1 and 65535

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station and perform a Standard Login.

Observable Results:
- Verify that if the DUT attempts to negotiate MaxConnections, it only does so in the leading connection of a session.
- Verify that the desired MaxConnections value falls within the required range.

Possible Problems: None.

Test #11.1 Initiator Name Target Name

Purpose: To verify that the DUT properly uses the InitiatorName and TargetName key=value pairs.

Reference: [RFC-7143] Section 6.3, 13.4, 13.5

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 13.4

TargetName – "The initiator of the TCP connection MUST provide this key to the remote endpoint in the first login request if the initiator is not establishing a Discovery session. The iSCSI Target Name specifies the worldwide unique name of the target."

[RFC-7143] Section 13.5

InitiatorName – "The initiator of the TCP connection MUST provide this key to the remote endpoint at the first Login of the Login Phase for every connection. The InitiatorName key enables the initiator to identify itself to the remote endpoint."

[RFC-7143] Section 6.3

"Neither the initiator nor the target should attempt to declare or negotiate a parameter more than once during login, except for responses to specific keys that explicitly allow repeated key declarations"

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.

Observable Results:

- Verify that the initial Login Request offered by the DUT contains both InitiatorName and TargetName keys.
- Verify that neither of these keys is renegotiated during the login.

Possible Problems: None.

Test #12.1 Marker Negotiation

Purpose: To verify that the DUT properly handles deprecated keys: OFMarker, IFMarker, OFMarkInt, IFMarkInt.

Reference: [RFC-7143] Section 13.25

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.25

> "This document obsoletes the following keys defined in [RFC3720]: IFMarker, OFMarker, OFMarkInt, and IFMarkInt. However, iSCSI implementations compliant to this document may still receive these obsoleted keys -- i.e., in a responder role -- in a text negotiation.
>
> When an IFMarker or OFMarker key is received, a compliant iSCSI implementation SHOULD respond with the constant "Reject" value. The implementation MAY alternatively respond with a "No" value. However, the implementation MUST NOT respond with a "NotUnderstood" value for either of these keys.
>
> When an IFMarkInt or OFMarkInt key is received, a compliant iSCSI implementation MUST respond with the constant "Reject" value. The implementation MUST NOT respond with a "NotUnderstood" value for either of these keys."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- If the initiator did not offer OFMarker, IFMarker, OFMarkInt, or IFMarkInt then transmit a response with the following key=value pairs: OFMarker=Yes, IFMarker=Yes, OFMarkInt range of 1~65535, IFMarkInt range of 1~65535.

Observable Results:

- Verify that the DUT does not offer these keys
- Verify that that DUT does not respond to these keys with a value of NotUnderstood
- Verify that the DUT responds to the IFMarker and OFMarker keys with a value of "Reject" or "No"

- Verify that the DUT responds to the IFMarkInt and OFMarkInt keys with a value of "Reject"

Possible Problems: None.

Test #13.1 Boolean Negotiation

Purpose: To verify that the DUT properly negotiates Immediate Data, InitialR2T, BiDiInitialR2T, DataPDUInOrder, DataSequenceInOrder.

Reference: [RFC-7143] Section 6.2.2, 13.10, 13.11, 13.18, 13.19

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6.2.2

> "For Boolean negotiations (i.e., keys taking the values Yes or No), the accepting party MUST answer with the same key and the result of the negotiation when the received value does not determine that result by itself. The last value transmitted becomes the negotiation result. The rules for selecting the value to answer with are expressed as Boolean functions of the value received, and the value that the accepting party would have selected if given a choice.
>
> Specifically, the two cases in which answers are OPTIONAL are:
>
> - The Boolean function is "AND" and the value "No" is received, the outcome of the negotiation is "No".
> - The Boolean function is "OR" and the value "Yes" is received, the outcome of the negotiation is "Yes".
>
> Responses are REQUIRED in all other cases, and the value chosen and sent by the acceptor becomes the outcome of the negotiation."

[RFC-7143] Section 13.10

> InitialR2T – "Result function is OR."

[RFC-7143] Section 13.11

> ImmediateData – "Result function is AND."

[RFC-7143] Section 13.18

> DataPDUInOrder – "Result function is OR."

[RFC-7143] Section 13.19

> DataSequenceInOrder – "Result function is OR."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit a response with the following key=value pairs: InitialR2T=No, Immediate Data=Yes, DataPDUInOrder=No, DataSequenceInOrder=No.

Observable Results:

- Verify that the DUT responds to the InitialR2T, Immediate Data, DataPDUInOrder, and DataSequenceInOrder keys during the Login phase. Responses are required in all cases.
- Verify that all values offered by the device in response begin with capital letters.

Possible Problems: None.

Test #13.2 Legal Boolean Negotiation

Purpose: To verify that the DUT properly recognizes errors in the negotiation of Immediate Data, InitialR2T, BiDiInitialR2T, DataPDUInOrder, DataSequenceInOrder.

Reference: [RFC-7143] Section 6.2.2, 13.10, 13.11, 13.18, 13.19

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6.2.2

> "For Boolean negotiations (i.e., keys taking the values Yes or No), the accepting party MUST answer with the same key and the result of the negotiation when the received value does not determine that result by itself. The last value transmitted becomes the negotiation result. The rules for selecting the value to answer with are expressed as Boolean functions of the value received, and the value that the accepting party would have selected if given a choice.
>
> Specifically, the two cases in which answers are OPTIONAL are:
>
> - The Boolean function is "AND" and the value "No" is received, the outcome of the negotiation is "No".
> - The Boolean function is "OR" and the value "Yes" is received, the outcome of the negotiation is "Yes".
>
> Responses are REQUIRED in all other cases, and the value chosen and sent by the acceptor becomes the outcome of the negotiation."

[RFC-7143] Section 13.10

> InitialR2T – "Result function is OR."

[RFC-7143] Section 13.11

> ImmediateData – "Result function is AND."

[RFC-7143] Section 13.18

> DataPDUInOrder – "Result function is OR."

[RFC-7143] Section 13.19

> DataSequenceInOrder – "Result function is OR."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login. In separate logins perform each of the following:
    - If the InitialR2T key is received with a value of Yes, do not return the key. Proceed through Login to the Full Feature Phase.
    - If the InitialR2T key is received with a value of No, return Yes. Proceed through Login to the Full Feature Phase.
    - If the DataPDUInOrder key is received with a value of Yes, do not return the key. Proceed through Login to the Full Feature Phase.
    - If the DataPDUInOrder key is received with a value of No, return Yes. Proceed through Login to the Full Feature Phase.
    - If the DataSequenceInOrder key is received with a value of Yes, do not return the key. Proceed through Login to the Full Feature Phase.
    - If the DataSequenceInOrder key is received with a value of No, return Yes. Proceed through Login to the Full Feature Phase.
    - If the Immediate Data key is received with a value of No, do not return the key. Proceed through login to Full Feature Phase.
    - If the Immediate Data key is received with a value of Yes, respond with No. Proceed through login to Full Feature Phase.

Observable Results:

- Verify that the DUT does NOT terminate the connection in any of these cases.
- Verify that the iSCSI initiator initiates SCSI traffic, to show that it has completed Login and entered Full Feature Phase.

Possible Problems: In each case, if the DUT does not offer the specified keys, the result of this test is “Not Supported”.

Test #13.3 Illegal Boolean Negotiation

Purpose: To verify that the DUT properly recognizes errors in the negotiation of Immediate Data, InitialR2T, BiDiInitialR2T, DataPDUInOrder, DataSequenceInOrder.

Reference: [RFC-7143] Section 6.2.2, 13.10, 13.11, 13.18, 13.19

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6.2.2

> "For Boolean negotiations (i.e., keys taking the values Yes or No), the accepting party MUST answer with the same key and the result of the negotiation when the received value does not determine that result by itself. The last value transmitted becomes the negotiation result. The rules for selecting the value to answer with are expressed as Boolean functions of the value received, and the value that the accepting party would have selected if given a choice.
>
> Specifically, the two cases in which answers are OPTIONAL are:
>
> - The Boolean function is "AND" and the value "No" is received, the outcome of the negotiation is "No".
> - The Boolean function is "OR" and the value "Yes" is received, the outcome of the negotiation is "Yes".
>
> Responses are REQUIRED in all other cases, and the value chosen and sent by the acceptor becomes the outcome of the negotiation."

[RFC-7143] Section 13.10

InitialR2T – "Result function is OR."

[RFC-7143] Section 13.11

ImmediateData – "Result function is AND."

[RFC-7143] Section 13.18

DataPDUInOrder – "Result function is OR."

[RFC-7143] Section 13.19

DataSequenceInOrder – "Result function is OR."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login. In separate logins perform each of the following:
    - If the InitialR2T key is received with a value of No, do not return the key. Proceed through Login to the Full Feature Phase.
    - If the InitialR2T key is received with a value of Yes, return No. Proceed through Login to the Full Feature Phase.
    - If the DataPDUInOrder key is received with a value of No, do not return the key. Proceed through Login to the Full Feature Phase.
    - If the DataPDUInOrder key is received with a value of Yes, return No. Proceed through Login to the Full Feature Phase.
    - If the DataSequenceInOrder key is received with a value of No, do not return the key. Proceed through Login to the Full Feature Phase.
    - If the DataSequenceInOrder key is received with a value of Yes, return No. Proceed through Login to the Full Feature Phase.
    - If the Immediate Data key is received with a value of Yes, do not return the key. Proceed through login to Full Feature Phase.
    - If the Immediate Data key is received with a value of No, respond with Yes. Proceed through login to Full Feature Phase.

Observable Results:

- Verify that the DUT detects a Negotiation Failure in each of these cases, and terminates the connection.

Possible Problems: In each case, if the DUT does not offer the specified keys, the result of this test is “Not Supported”.

Test #14.1 MaxRecvDataSegmentLength

Purpose: To verify that the DUT properly recognizes the MaxRecvDataSegmentLength key=value pair.

Reference: [RFC-7143] Section 13.12

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.2

MaxRecvDataSegmentLength is defined as a numerical value between 512 and 2^{24} -1. The default is 8192 bytes

"The initiator or target declares the maximum data segment length in bytes it can receive in an iSCSI PDU.

The transmitter (initiator or target) is required to send PDUs with a data segment that does not exceed MaxRecvDataSegmentLength of the receiver."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit a response with MaxRecvDataSegmentLength 4096.

Observable Results:
- Verify that the DUT supports the MaxRecvDataSegmentLength key during the Login phase, (i.e. no error should be generated.)

Possible Problems: None.

Test #15.1 MaxBurstLength

Purpose: To verify that the DUT properly negotiates the MaxBurstLength key=value pair.

Reference: [RFC-7143] Section 13.13

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.13

MaxBurstLength is defined as a numerical value between 512 and 2^{24}-1. The default value is 262144. It may only be used in the Leading Login of a session.

"The initiator and target negotiate the maximum SCSI data payload in bytes in a Data-In or a solicited Data-Out iSCSI sequence."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit a response with the value of the MaxBurstLength key greater than the default value for FirstBurstLength, but less than the default value for MaxBurstLength.

Observable Results:
- Verify that the MaxBurstLength key is responded to properly by the device under test. Verify the response value is less than or equal to the requested value.

Possible Problems: None.

Test #16.1 FirstBurstLength Less Than MaxBurstLength

Purpose: To verify that the DUT properly negotiates the FirstBurstLength key=value pair.

Reference: [RFC-7143] Section 13.14

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.14

FirstBurstLength is defined as a numerical value between 512 and $2^{24} – 1$. It may only be used in the leading login of a session.

"The initiator and target negotiate the maximum amount in bytes of unsolicited data an iSCSI initiator may send to the target during the execution of a single command.

FirstBurstLength MUST NOT exceed MaxBurstLength."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit a response with the FirstBurstLength key with a value less than the current value for MaxBurstLength.

Observable Results:
- Verify that the FirstBurstLength key is responded to properly by the device under test. Verify the value requested falls under any value negotiated for MaxBurstLength.

Possible Problems: None.

Test #16.2 FirstBurstLength Greater Than MaxBurstLength

Purpose: To verify that the DUT properly negotiates the FirstBurstLength key=value pair.

Reference: [RFC-7143] Section 13.14

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.14

FirstBurstLength is defined as a numerical value between 512 and $2^{24} - 1$. It may only be used in the leading login of a session.

"The initiator and target negotiate the maximum amount in bytes of unsolicited data an iSCSI initiator may send to the target during the execution of a single command.

FirstBurstLength MUST NOT exceed MaxBurstLength."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Negotiate Immediate Data and InitialR2T such that FirstBurstLength is relevant.
- Transmit a response with the FirstBurstLength key, greater than the default MaxBurstLength key.

Observable Results:
- Verify that the FirstBurstLength key is either rejected by the DUT and the DUT may then disconnect, or that the DUT responds with a value of FirstBurstLength that is less than the current MaxBurstLength, which may also be the default.

Possible Problems: If the DUT does not support values for Immediate Data and InitialR2T that make FirstBurstLength relevant, the result of this test is "Not Supported".

Test #16.3 FirstBurstLength Offered by DUT

Purpose: To verify that the DUT properly negotiates the FirstBurstLength key=value pair.

Reference: [RFC-7143] Section 13.14

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.14

FirstBurstLength is defined as a numerical value between 512 and $2^{24} – 1$. It may only be used in the leading login of a session.

"The initiator and target negotiate the maximum amount in bytes of unsolicited data an iSCSI initiator may send to the target during the execution of a single command.

FirstBurstLength MUST NOT exceed MaxBurstLength."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station and perform a Standard Login.

Observable Results:
- If the device offers the FirstBurstLength key, verify that it is not greater than the current value of MaxBurstLength, which may be the default.
- Verify that a device, which uses the FirstBurstLength key, only does so in the leading login of a session.

Possible Problems: None.

Test #17.1 DefaultTime2Retain

Purpose: To verify that the DUT properly negotiates the DefaultTime2Retain key=value pair.

Reference: [RFC-7143] Section 13.16

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.16

DefaultTime2Retain is defined as a numerical value between 0 and 3600. It may only be used in the leading connection of a session.

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit a response with the DefaultTime2Retain key with a value of 300.

Observable Results:

- If the DUT does not originate the DefaultTime2Retain key, verify that it responds to the DefaultTime2Wait key offered by the Testing Station with a value less than or equal to the value offered by the Testing Station.
- If the DUT is the originator of the DefaultTime2Retain key that it presents a range of values with a ~ between 0 and 3600.

Possible Problems: None.

Test #18.1 DefaultTime2Wait

Purpose: To verify that the DUT properly negotiates the DefaultTime2Wait key=value pair.

Reference: [RFC-7143] Section 13.15

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.15

DefaultTime2Wait is defined as a numerical value between 0 and 3600. It may only be used in the leading connection of a session.

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit a response with the DefaultTime2Wait key with a value of 300.

Observable Results:

- If the DUT does not originate the DefaultTime2Wait key, verify that it responds to the DefaultTime2Wait key offered by the Testing Station with a value less than or equal to the value offered by the Testing Station.
- If the DUT is the originator of the DefaultTime2Wait key that it presents a range of values with a ~ between 0 and 3600.

Possible Problems: None.

Test #19.1 MaxOutstandingR2T

Purpose: To verify that the DUT properly negotiates the MaxOutstandingR2T key=value pair.

Reference: [RFC-7143] Section 13.17

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.17

MaxOutstandingR2T is defined as a numerical value between 1 and 65535. It may only be used in the leading login of a session.

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.
- Transmit a response with the MaxOutstandingR2T key equal to 4

Observable Results:

- Verify that a device, which originates the MaxOutstandingR2T key, only does so in the leading connection of a session, and that the values it presents fall between 1 - 65535.
- Verify that a device, which does not originate the MaxOutstandingR2T key, responds with a value less than or equal to the value offered by the Testing Station.

Possible Problems: None.

Test #20.1 ErrorRecoveryLevel

Purpose: To verify that the DUT properly negotiates the ErrorRecoveryLevel key=value pair.

Reference: [RFC-7143] Section 13.20

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.20

ErrorRecoveryLevel is defined as a numerical value between 0 and 2. The minimum of the two values is selected.

"The initiator and target negotiate the recovery level supported.

Recovery levels represent a combination of recovery capabilities. Each recovery level includes all the capabilities of the lower recovery levels and adds some new ones to them."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the DUT to connect to the Testing Station and perform a Standard Login.
- Transmit a response with the ErrorRecoveryLevel=2 key.

Observable Results:
- If the DUT offers ErrorRecoveryLevel, verify that it is 0, 1, or 2.
- Otherwise, verify that the DUT responds to the ErrorRecoveryLevel key with 0, 1, or 2.

Possible Problems: None.

Test #21.1 SessionType

Purpose: To verify that the DUT properly recognizes the SessionType key=value pair.

Reference: [RFC-7143] Section 13.21

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.21

SessionType - "The Initiator indicates the type of session it wants to create. The target can either accept it or reject it.

A Discovery session indicates to the Target that the only purpose of this Session is discovery. The only requests a target accepts in this type of session are a text request with a SendTargets key and a logout request with reason "close the session".

The Discovery session implies MaxConnections = 1 and overrides both the default and an explicit setting."

The SessionType key may only be used on the leading login of a connection.

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and perform a Standard Login.

Observable Results:

- Verify that if the DUT uses the SessionType key, only does so in the leading connection of a session, and that formats the key=value pair properly.
- Verify that if the SessionType=Normal, that the TargetName key is also included in the login request.

Possible Problems: None.

Test #22.1 AuthMethod

Purpose: To verify that the DUT supports the AuthMethod value of CHAP.

Reference: [RFC-7143] Section 12.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 12.1

> "The authentication methods that can be used . . . are vendor-unique methods or those listed": KRB5, SPKM1, SPKM2, SRP, CHAP, None.
>
> "The initiator and target MUST implement CHAP. All other authentication methods are OPTIONAL."

[RFC-7143] Section 15

> "IANA has marked as obsolete the values 4 and 5 for SPKM1 and SPKM2, respectively, in the "iSCSI authentication methods" subregistry of the "Internet Small Computer System Interface (iSCSI) Parameters" set of registries. . . . This document obsoletes the SPKM1 and SPKM2 key values for the AuthMethod text key."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Wait for the initiator to connect to the Testing Station.
- In the Security Negotiation Stage, offer the following key=value pair: AuthMethod = KRB5, SPKM1, SPKM2, SRP, CHAP.

Observable Results:
- If the DUT does not originate AuthMethod, verify that the DUT chooses a valid value from the list offered by the Testing Station (i.e., does not choose SPKM1 or SPKM2)
- If the DUT originates AuthMethod, verify that CHAP is included in the list and that SPKM1 and SPKM2 are not in the list.

Possible Problems: None.

Test #23.1 TargetPortalGroupTag

Purpose: To verify that the DUT requires the presence of the TargetPortalGroupTag in the first Login Response it receives.

Reference: [RFC-7143] Section 6.3.1, 13.9

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6.3.1

> "During the Login Phase the iSCSI target MUST return the TargetPortalGroupTag key with the first Login Response PDU with which it is allowed to do so (i.e., the first Login Response issued after the first Login Request with the C bit set to 0)."

[RFC-7143] Section 13.9

> "The target portal group tag is a 16-bit binary-value that uniquely identifies a portal group within an iSCSI target node. This key carries the value of the tag of the portal group that is servicing the Login request. The iSCSI target returns this key to the initiator in the Login Response PDU to the first Login Request PDU that has the C bit set to 0 when the TargetName is given by the initiator."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Wait for the initiator to connect to the Testing Station and start a Normal Session.
- If the first Login Request from the DUT has C=0, then the first Login Response from the Testing Station should not offer the TargetPortalGroupTag key.
- Wait for the DUT to send a second Login Request, or drop the connection.
- In the second Login Response the Testing Station should offer the key=value pair TargetPortalGroupTag=ABCD.

Observable Results:

- Verify that the DUT requires the TargetPortalGroupTag to be included in the first Login Response PDU. The DUT should drop the connection when this key is not included in the first Login Response.

Possible Problems: None.

Test #24.1 C bit with Small Logical Text Data Segment

Purpose: To verify that the DUT properly handles a Login Response PDU with the C bit set.

Reference: [RFC-7143] Section 6, 6.1, 6.2, 11.13.7

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6

> "Since some key=value pairs may not fit entirely in a single PDU, the C (Continue) bit is used (both in Login and Text) to indicate that "more follows"."

[RFC-7143] Section 6.1

> "Key=value pairs may span PDU boundaries. An initiator or target that sends partial key=value text within a PDU indicates that more text follows by setting the C bit in the Text or Login Request or Text or Login Response to 1. Data segments in a series of PDUs that have the C bit set to 1 and end with a PDU that have the C bit set to 0, or include a single PDU that has the C bit set to 0 have to be considered as forming a single logical-text-data-segment (LTDS)."

[RFC-7143] Section 6.2

> "As negotiation text may span PDU boundaries, a Text or Login Request or Text or Login Response PDU that have the C bit set to 1 MUST NOT have the F/T bit set to 1."
>
> "An initiator receiving a Text or Login Response with the C bit set to 1 MUST answer with a Text or Login Request with no data segment (DataSegmentLength=0)."

[RFC-7143] Section 11.13.7

> "When set to 1, the C bit indicates that the text (set of key=value pairs) in this Login Response is not complete (it will be continued on subsequent Login Responses); otherwise, it indicates that this Login Response ends a set of key=value pairs. A Login Response with the C bit set to 1 MUST have the T bit set to 0."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Response to the DUT, with the C bit =1. T bit = 0, and the following keys: X-cbit.ioliscsilab.test-1 = 255 bytes of random data. The final key =value pair in this request should be 'MaxRecvDataSegment' then the end of the data segment.

- Transmit a second Text Request to the DUT with the C bit = 0, T bit = 1, and the final portion of the request: 'Length=512'.
- Proceed to the Full Feature Phase.
- Wait for a WRITE command from the DUT and transmit R2T.

Observable Results:

- The DUT should transmit 'NotUnderstood' to the vendor specific keys. The DUT should not disconnect.
- Verify that the Login Request transmitted after receiving the Login Response with the C bit set to 1, has no data segment.
- Verify that the DUT adheres to the MaxRecvDataSegmentLength declared by the Testing Station in the Full Feature Phase.

Possible Problems: None.

Test #24.2 C bit with Large Logical Text Data Segment

Purpose: To verify that the DUT properly handles a Login Response PDU with the C bit set.

Reference: [RFC-7143] Section 6, 6.1, 6.2, 11.13.7

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6

> "Since some key=value pairs may not fit entirely in a single PDU, the C (Continue) bit is used (both in Login and Text) to indicate that "more follows"."

[RFC-7143] Section 6.1

> "Key=value pairs may span PDU boundaries. An initiator or target that sends partial key=value text within a PDU indicates that more text follows by setting the C bit in the Text or Login Request or Text or Login Response to 1. Data segments in a series of PDUs that have the C bit set to 1 and end with a PDU that have the C bit set to 0, or include a single PDU that has the C bit set to 0 have to be considered as forming a single logical-text-data-segment (LTDS)."

[RFC-7143] Section 6.2

> "As negotiation text may span PDU boundaries, a Text or Login Request or Text or Login Response PDU that have the C bit set to 1 MUST NOT have the F/T bit set to 1."
>
> "An initiator receiving a Text or Login Response with the C bit set to 1 MUST answer with a Text or Login Request with no data segment (DataSegmentLength=0)."

[RFC-7143] Section 11.13.7

> "When set to 1, the C bit indicates that the text (set of key=value pairs) in this Login Response is not complete (it will be continued on subsequent Login Responses); otherwise, it indicates that this Login Response ends a set of key=value pairs. A Login Response with the C bit set to 1 MUST have the T bit set to 0."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Response to the DUT, with the C bit =1. T bit = 0, and the following keys: X-cbit.ioliscsilab.test-n = 255 bytes of random data. Keys with values for 'n' = 1 -

32 should be included in this request, up to 8192 bytes. The final key =value pair in this request should be 'MaxRecvDataSegment' then the end of the data segment.
- Transmit a second Text Request to the DUT with the C bit = 0, T bit = 1, and the final portion of the request: 'Length=512'.
- Proceed to the Full Feature Phase.
- Wait for a WRITE command from the DUT and transmit R2T.

Observable Results:
- The DUT should transmit 'NotUnderstood' to the vendor specific keys. The DUT should not disconnect.
- Verify that the Login Request transmitted after receiving the Login Response with the C bit set to 1, has no data segment.
- Verify that the DUT adheres to the MaxRecvDataSegmentLength declared by the Testing Station in the Full Feature Phase.

Possible Problems: The DUT is not required to support receiving more than 8192 bytes in a negotiation sequence. If the DUT does not support this, the result of this test is “Not Supported”.

Test #25.1 Redirect

Purpose: To verify that the DUT properly handles a Target redirection.

Reference: [RFC-7143] Section 11.13.5

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 11.13.5

In a Login Response PDU, the Status "indicates the execution status of the Login Phase. The status includes:

- Status-Class
- Status-Detail.

A Status-Class of 0 indicates success.

A non-zero Status-Class indicates an exception."

Status-Class 1: Redirection: "indicates that the initiator must take further action to complete the request. This is usually due to the target moving to a different address. All of the redirection status class responses MUST return one or more text key parameters of the type "TargetAddress", which indicates the target's new address."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and wait for the first Login Request from the DUT.
- Transmit a Login Response to the DUT with a Status Class = 0x01 and a Status Detail =0x01 to indicate that the target has changed addresses permanently. The Testing Station should include a new address with the TargetAddress key=value pair.

Observable Results:

- Verify that the DUT disconnects and then reconnects to the new TargetAddress.

Possible Problems: None.

Test #26.1 Errors Invalid Keys

Purpose: To verify that the DUT recognizes keys that are invalid for a target to transmit.

Reference: [RFC-7143] Section 6.2, 7.13, 13

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: November 3, 2015

Discussion:

[RFC-7143] Section 13

> Each Login/Text Operational Keys definition includes "Senders" (initiator or target or both). If a target were to transmit a key not allowed for targets, this would indicate a protocol error.

[RFC-7143] Section 7.13

> "All violations of iSCSI PDU exchange sequences specified in this document are also protocol errors. This category of errors can only be addressed by fixing the implementations; iSCSI defines Reject and response codes to enable this."

[RFC-7143] Section 6.2

> "An iSCSI implementation MUST comprehend all text keys defined in this document. . . . All keys in this document MUST be supported by iSCSI initiators and targets when used as specified here. If used as specified, these keys MUST NOT be answered with NotUnderstood."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Response with the following keys=value pairs: InitiatorName=iqn.2002.UNH.EDU, InitiatorAlias=UNHIOL.

Observable Results:

- The DUT should respond with key=Reject or key=NotUnderstood.
- The DUT may also disconnect due to a protocol error since the target is not allowed to send these keys according to [RFC-7143] Section sections 13.5 and 13.7.

Possible Problems: None.

Test #26.2.1 Errors X Keys

Purpose: To verify that the DUT properly responds to received X keys.

Reference: [RFC-7143] Section 6.2

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.2

> "Any key not understood by the acceptor may be ignored by the acceptor without affecting the basic function. However, the answer for a key that is not understood MUST be key=NotUnderstood."
>
> "Implementers may introduce new private keys by prefixing them with X- followed by their (reverse) domain name, or with new public keys registered with IANA."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Response with the following keys=value pair: X-edu.unh.iol-extension-key-1=test.

Observable Results:

- The DUT should answer the received keys with the value 'NotUnderstood'.

Possible Problems: None.

Test #26.2.2 Errors X Key Name Too Large

Purpose: To verify that the DUT properly responds to received X keys.

Reference: [RFC-7143] Section 6.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.1

> Key names are defined as standard-labels: "A string of one or more characters that consists of letters, digits, dot, minus, plus, commercial at, or underscore. A standard-label MUST begin with a capital letter and must not exceed 63 characters."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Response with the following keys=value pair: X-edu.unh.iol-extension-key-which-is-clearly-longer-than-it-ought-to-be-1=test (73 characters).

Observable Results:
- Verify that the DUT disconnects due to a protocol error since the key name is greater than 63 characters.
- Verify that the DUT does not respond with a value of Reject or NotUnderstood, as it is a protocol error to transmit the key.
- Verify that the DUT does not respond with a truncated version of the key.

Possible Problems: None.

Test #26.3.1 Errors Big Values Boolean Value

Purpose: To verify that the DUT properly recognizes values that exceed the 255 byte limit for values.

Reference: [RFC-7143] Section 6.1, 6.2.2

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Sections 6.1

> "If not otherwise specified, the maximum length of a simple-value (not its encoded representation) is 255 bytes not including the delimiter (comma or zero byte)."

[RFC-7143] Section 6.2.2

> "For simple-value negotiations, the accepting party MUST answer with the same key. The value it selects becomes the negotiation result.
>
> Proposing a value not admissible (e.g., not within the specified bounds) MAY be answered with the constant "Reject"; otherwise, the acceptor MUST select an admissible value."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Response with the following keys=value pair: ImmediateData=No.

Observable Results:

- The DUT should reject the received key. The DUT may also select an admissible value.
- The DUT may disconnect due to a protocol error since the length of the key value is greater than 255 bytes.

Possible Problems: None.

Test #26.3.2 Errors Big Values Declared Value (Informative)

Purpose: To verify that the DUT properly recognizes values that exceed the 255 byte limit for values. This test is meant to be informative only.

Reference: [RFC-7143] Section 6.1, 7.10, 13.6

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6.1

> "If not otherwise specified, the maximum length of a simple-value (not its encoded representation) is 255 bytes not including the delimiter (comma or zero byte)."

[RFC-7143] Section 7.10

> "Unless this document requires it, an iSCSI implementation is not required to do an exhaustive protocol conformance check on an incoming iSCSI PDU. The iSCSI implementation in particular is not required to double-check the remote iSCSI implementation's conformance to protocol requirements."

[RFC-7143] Section 13.6

> The TargetAlias key is defined as Declarative, indicating that no response is necessary.

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Response with the following keys=value pair: TargetAlias = WickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTarget

Observable Results:

- The DUT should reject the received key.
- The DUT may disconnect due to a protocol error.

Possible Problems: This is an informative test. It cannot be verified whether a device is using an invalid value when it does not terminate the connection during this test. The possibility exists

that a iSCSI device may only read 255 bytes of data since that is all that this valid, and may never detect that an invalid value is being used. The integrity checking rules defined in [RFC-7143] Section 6.2 do not apply here.

Test #26.4 Errors Inquire Value

Purpose: To verify that the DUT properly recognizes invalid values.

Reference: [RFC-7143] Section 6.2

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.2

"The general format of text negotiation is:

Proposer-> <key>=<valuex>

Acceptor-> <key>={<valuey>|NotUnderstood|Irrelevant|Reject}"

Note that the '?' inquire value is not an allowed response.

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Request with the following keys=value pair: MaxConnections=?

Observable Results:
- The DUT should reject the received key. The DUT may also select an admissible value.
- The DUT may disconnect due to a protocol error.

Possible Problems: None.

Test #27.1 Irrelevant Keys

Purpose: To verify that the DUT properly handles keys which are irrelevant during a Discovery Session.

Reference: [RFC-7143] Section 6.2, 13.2, 13.10, 13.11, 13.13, 13.14, 13.17, 13.18, 13.19, 13.23

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:

[RFC-7143] Section 6.2

> "If a specific key is not relevant for the current negotiation, the acceptor may answer with the constant "Irrelevant" for all types of negotiations. However, the negotiation is not considered to have failed if the answer is "Irrelevant"."

[RFC-7143] Section 13.2, 13.10, 13.11, 13.13, 13.14, 13.17, 13.18, 13.19, 13.23

> MaxConnections, InitialR2T, Immediate Data, MaxBurstLength, FirstBurstLength, MaxOutstandingR2T, DataPDUInOrder, DataSequenceInOrder and TaskReporting – "Irrelevant when: SessionType=Discovery"

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT. Allow the DUT to start a Discovery Session.
- After receiving the first Login Request from the DUT with SessionType=Discovery, the Testing Station should transmit a Login Response with the following keys, each with a valid value: MaxConnections, InitialR2T, Immediate Data, MaxBurstLength, FirstBurstLength, MaxOutstandingR2T, DataPDUInOrder, DataSequenceInOrder and TaskReporting.

Observable Results:

- The DUT should not drop the connection. The DUT is expected to respond to the keys 'Irrelevant'.

Possible Problems: None.

Test #28.1 Error Recovery for Discovery Sessions

Purpose: To verify that the DUT properly handles the ErrorRecoveryLevel key in a discovery session.

Reference: [RFC-7143] Section 7.4.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 7.4.1

> "The negotiation of the key ErrorRecoveryLevel is not required for Discovery sessions -- i.e., for sessions that negotiated "SessionType=Discovery" -- because the default value of 0 is necessary and sufficient for Discovery sessions. It is, however, possible that some legacy iSCSI implementations might attempt to negotiate the ErrorRecoveryLevel key on Discovery sessions. When such a negotiation attempt is made by the remote side, a compliant iSCSI implementation MUST propose a value of 0 (zero) in response."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT. Allow the DUT to start a Discovery Session.
- After receiving the first Login Request from the DUT with SessionType=Discovery, the Testing Station should transmit a Login Response with ErrorRecoveryLevel=1.

Observable Results:

- The DUT should not initiate the negotiation of ErrorRecoveryLevel greater than 0.
- The DUT should negotiate ErrorRecoveryLevel=0.

Possible Problems: None.

Test #29.1 NotUnderstood for Required Keys

Purpose: To verify that the DUT treats a response of NotUnderstood as a protocol error for keys defined in RFC7143.

Reference: [RFC-7143] Section 6.2

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.2

> "An iSCSI implementation MUST comprehend all text keys defined in this document. Returning a NotUnderstood response on any of these text keys therefore MUST be considered a protocol error and handled accordingly."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- After receiving the first Login Request from the DUT, the Testing Station should transmit a Login Response with TargetName=NotUnderstood.

Observable Results:

- Verify that the DUT disconnects due to a protocol error.

Possible Problems: None.

Test #30.1 TaskReporting

Purpose: To verify that the DUT properly negotiates the TaskReporting key=value pair.

Reference: [RFC-7143] Section 6.2, 13.23

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 13.23

TaskReporting – "This key is used to negotiate the task completion reporting semantics from the iSCSI target. . . . Whenever this key is negotiated, at least the RFC3720 and ResponseFence values MUST be offered as options by the negotiation originator."

A value of FastAbort may also be negotiated.

[RFC-7143] Section 6.2

"An iSCSI implementation MUST comprehend all text keys defined in this document. Returning a NotUnderstood response on any of these text keys therefore MUST be considered a protocol error and handled accordingly."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- After receiving the first Login Request from the DUT, if the DUT did not offer the TaskReporting key, the Test Station should send a Login Response with TaskReporting=RFC3720,ResponseFence,FastAbort. If the DUT did offer the TaskReporting key, the Test Station should send a Login Response with TaskReporting=RFC3720.

Observable Results:

- If the DUT originated TaskReporting; verify that the DUT offered RFC3720 and ResponseFence.
- If the DUT did not originate TaskReporting; verify that the DUT chooses a value from the list offered by the TestStation.

Possible Problems: None.

Test #31.1 iSCSIProtocolLevel (Informative)

Purpose: To verify that the DUT properly negotiates the iSCSIProtocolLevel key=value pair.

Reference: [RFC-7143] Section 13.24, [RFC-7144] Section 7.1.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 23, 2015

Discussion:

[RFC-7143] Section 13.24

iSCSIProtocolLevel – "The iSCSIProtocolLevel associated with this document is "1". As a responder or an originator in a negotiation of this key, an iSCSI implementation compliant to this document alone, without any future protocol extensions, MUST use this value as defined by [RFC7144]."

[RFC-7144] Section 7.1.1

iSCSIProtocolLevel – "This key is used to negotiate the use of iSCSI features that require different levels of protocol support (e.g., PDU formats, end-node semantics) for proper operation."

"An iSCSIProtocolLevel key negotiated to "2" is required to enable use of features defined in this RFC."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and begin a standard login.
- After receiving the first Login Request from the DUT, the Test Station should send a Login Response with iSCSIProtocolLevel=1.

Observable Results:

- Verify that the DUT offers or responds with iSCSIProtocolLevel=1 or iSCSIProtocolLevel=2.

Possible Problems: This is an informative test. A DUT that does not claim compliance to RFC 7143 may respond with iSCSIProtocolLevel=0. It may otherwise ignore the key or respond with a value of "NotUnderstood". This behavior does not violate the requirements of RFC 7143.

Test #32.1 Public Extension Keys (Informative)

Purpose: To verify that the DUT no longer uses the X#, Y#, or Z# prefixes for new public keys, new digest extensions, or new authentication method extensions.

Reference: [RFC-7143] Section 6.2, 12.1, 13.1

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 23, 2015

Discussion:

[RFC-7143] Section 6.2

"Implementers may introduce . . . new public keys registered with IANA."

"Each new public key in the course of standardization MUST define the acceptable responses to the key, including NotUnderstood as appropriate. Unlike [RFC3720], note that this document prohibits the X# prefix for new public keys. Based on iSCSI implementation experience, we know that there is no longer a need for a standard name prefix for keys that allow a NotUnderstood response. Note that NotUnderstood will generally have to be allowed for new public keys for backwards compatibility, as well as for private X- keys. Thus, the name prefix "X#" in new public key-names does not carry any significance. To avoid confusion, new public key-names MUST NOT begin with an "X#" prefix."

[RFC-7143] Section 12.1

"New public extensions for authentication methods MUST NOT use the *Z#* name prefix."

[RFC-7143] Section 13.1

"New public keys must be registered with IANA using the IETF Review process ([RFC5226]). New public extensions for digests MUST NOT use the Y# name prefix."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:

- Connect the Testing Station to the DUT and complete a standard login.

Observable Results:

- Verify that the DUT does not offer keys with the "X#" prefix, the "Y#" prefix, or the "#Z" prefix, with the exception of X#NodeArchitecture, which was registered with the IANA prior to the release of RFC 7143.

Possible Problems: A reliable way of performing this test is not currently available.

Test #33.1 Receive Limit During Login

Purpose: To verify that the DUT properly handles a Login Response PDU with more than 8K of data attached.

Reference: [RFC-7143] Section 6.1, 5.2, 10.13.7

Resource Requirements: A Test Generator tool capable of producing iSCSI PDUs and transporting them over a TCP connection.

Last Modification: October 13, 2015

Discussion:
[RFC-7143] Section 6.1

> "Any iSCSI target or initiator MUST support receiving at least 8192 bytes of key=value data in a negotiation sequence. When proposing or accepting authentication methods that explicitly require support for very long authentication items, the initiator and target MUST support receiving of at least 64 kilobytes of key=value data."

Test Setup: The DUT and Test Station pair should be able to make a TCP connection.

Procedure:
- Connect the Testing Station to the DUT and begin a standard login.
- Transmit a Login Response to the DUT, with the following keys: X-ioliscsilab.test-n = 255 bytes of random data. Keys with values for 'n' = 1 - 32 should be included in this request, up to 8192 bytes. Also, MaxRecvDataSegmentLength=512 should be included in the request.
- Proceed to the Full Feature Phase.
- Wait for a WRITE command from the DUT for more than 512 bytes.

Observable Results:
- The DUT should transmit 'NotUnderstood' to the vendor specific keys. The DUT should not disconnect.
- The DUT should not disconnect upon receiving the Login Response.
- The DUT should use the MaxRecvDataSegmentLength value negotiated during Login.

Possible Problems: A reliable way of performing this test is not currently available.