UNH IOL iSCSI CONSORTIUM

Login Phase Test Suite for iSCSI Targets
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 29, 2005 (Version 1.0) FINAL RELEASE
Les Peabody: Test Suite updated to match final RFC 3720 standard.

[3] April 11, 2006 (Version 1.1) FINAL RELEASE
David Woolf: Corrected test 19.3.2.

[4] January 5, 2007 (Version 1.2) FINAL RELEASE
Aaron Bascom: Changed title page.

[5] March 25, 2008 (Version 2.0) DRAFT RELEASE
Ethan Burns: Test Suite updated to match final RFC 5048 standard.
Updated: 21.1
Added: 22.1, 23.1 and 24.1

[6] March 13, 2009 (Version 2.1) FINAL RELEASE
Mark Niemeyer: Updated tests 21.1 and 23.1.
Renamed tests to have unique names.

[7] September 3, 2009 (Version 2.2) DRAFT RELEASE
Patrick MacArthur: Updated tests 3.1, 6.1-6.5, 8.1, 9.1, 16.3, 19.1, 19.3.1, 19.3.2, and 19.4.
Added "Additional Acronyms and Abbreviations" section.

[8] July 6, 2015(Version 2.3) FINAL RELEASE
Aaron Morneau: Added test 12.3
Updated 12.2

[9] September 8, 2015 (Version 2.4) FINAL RELEASE
Andrew Johnson: Updated test suite with new manager and updated fingerprint information.

[10] November 3, 2015 (Version 3.0) FINAL RELEASE
Andrew Johnson: Updated references to RFC7143
Fixed broken headings
Amy Davies: Added RFC 7144 to References
Updated reference formatting
Updated address
Removed Digital Signature information.
Updated wording of Possible Problems to new "Not Supported" wording
Modified test #15.1 for deprecated keys
Modified test #19.2.2 to not allow invalid key to be transmitted in response
Modified test #24.1 to remove Possible Problem for unsupported TaskReporting key.
Added informative test #25.1 for iSCSIProtocolLevel key
Added test #26.1 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.

Aaron Bascom	UNH InterOperability Laboratory
Ethan Burns	UNH InterOperability Laboratory
Patrick MacArthur	UNH InterOperability Laboratory
Mark Niemeyer	UNH InterOperability Laboratory
Les Peabody	UNH InterOperability Laboratory
David Woolf	UNH InterOperability Laboratory
Aaron Morneau	UNH InterOperability Laboratory
Andrew Johnson	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 Full Feature Phase functionality of their iSCSI targets.

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 which 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 may also be 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

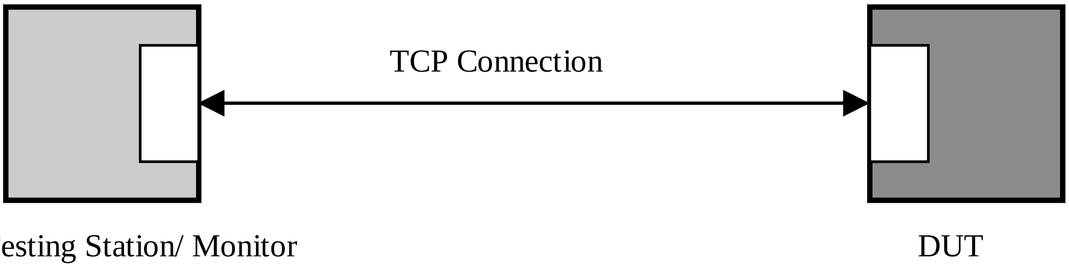
The acronyms and abbreviations defined here supplement the acronyms defined in IETF RFC 7143 section 2.1 and may be used in this document.

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:



GROUP 1: LOGIN PHASE FOR TARGETS

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, UNH InterOperability Lab (kerry.munson@iol.unh.edu).

Test #1.1: Standard Login Key Negotiation

Purpose: To verify that the DUT properly uses the following: Opcode, TSIH, CID, CmdSN, StatSN, TargetPortalGroupTag, InitialR2T, Immediate Data, MaxRecvDataSegmentLength, MaxBurstSize, FirstBurstSize, DefaultTime2Wait, DefaultTime2Retain, MaxOutstandingR2T, DataPDUInOrder, DataSequenceInOrder, ErrorRecoveryLevel. To verify that the DUT includes all pertinent information in its Final Response.

Reference [RFC-7143] Section 11.13, 11.13.3, 11.12.7, 11.12.8, 11.13.4, 13.9, 13.10, 13.11, 13.12, 13.13, 13.14, 13.15, 13.16, 13.17, 13.18, 13.19, 13.20, 6.3

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

Last Modification: October 26, 2015

Discussion:
[RFC-7143] Section 11.13

> The opcode for an iSCSI login response is 0x23.

[RFC-7143] Section 11.13.3

> “The TSIH is the target-assigned session-identifying handle. Its internal format and content are not defined by this protocol, except for the value 0, which is reserved. 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.”

[RFC-7143] Section 11.12.7

> “The CID provides a unique ID for this connection within the session.
>
> All Login requests within the Login phase MUST carry the same CID.
>
> The target MUST use the value presented with the first Login Request.”

[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.”

[RFC-7143] Section 11.13.4

> “For the first Login Response (the response to the first Login Request), [the StatSN] 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.”

[RFC-7143] Section 13.9

“The target portal group tag key 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.”

[RFC-7143] Section 13.10

“The InitialR2T key is used to turn off the default use of R2T.”

This parameter can only be used in the leading connection.

[RFC-7143] Section 13.11

ImmediateData- “The initiator and target negotiate support for immediate data.”

This parameter can only be used in the leading connection.

[RFC-7143] Section 13.12

MaxRecvDataSegmentLength - “The initiator or target declares the maximum data segment length in bytes it can receive in an iSCSI PDU.”

MaxRecvDataSegmentLength is defined as a number between 512 and 2^{24}-1.

[RFC-7143] Section 13.13

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

The MaxBurstLength key can only be used in the Leading Login of a connection and must be between 512 and 2^{24} -1.

[RFC-7143] Section 13.14

FirstBurstLength – “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.”

The FirstBurstLength key can only be used in the leading login of a session and must fall in a range between 512 and 2^{24}-1

[RFC-7143] Section 13.15 and 13.16

The DefaultTime2Wait and DefaultTime2Retain can only be used in the leading connection and must be a number from 0 - 3600.

[RFC-7143] Section 13.17

The MaxOutstandingR2T key can only be used in the leading login of a connection and must be a number from 1 - 65535.

[RFC-7143] Section 13.18 and 13.19

The DataPDUInOrder and DataSequenceInOrder key have a Yes|No value and can only be used in the Leading Login of a connection.

[RFC-7143] Section 13.20

The ErrorRecoveryLevel key can only be used in the Leading Login of a connection and must have a value between 0 and 2. Both initiator and target send this key. The minimum of the two values is selected.

[RFC-7143] Section 6.3

"The Login Phase sequence of requests and responses proceeds as follows:

- Login initial request
- Login partial response (optional)
- More Login Requests and Responses (optional)
- Login Final-Response (mandatory)"

"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.

The Login Requests and Responses contain a field (CSG) that indicates the current negotiation stage (SecurityNegotiation or LoginOperationalNegotiation). 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:

- Connect the Testing Station to the Target.
- The Testing Station should attempt to perform a standard login with the target. Use a value of 123 for CmdSN.
- Attempt to negotiate the following keys: InitialR2T, ImmediateData, MaxRecvDataSegmentLength, MaxBurstLength, FirstBurstLength, DefaultTime2Retain, DefaultTime2Wait, MaxOutstandingR2T, DataPDUInOrder, DataSequenceInOrder, ErrorRecoveryLevel.

Observable Results:

- Verify that in each login response that the DUT uses the 0x23 command code.
- Verify that the target sets the TSIH field only in the final login response, (i.e. the TSIH equals 0 in every response but the final login response). Verify that it is formatted properly.
- Verify that the target uses the CID provided in the first Login Request
- Verify that a target uses the provided value for CmdSN by checking ExpCmdSN.
- Verify that a target sets StatSN and increments it with each Login response.
- Verify that the DUT includes the TargetPortalGroupTag key=value pair in its first Login Response PDU.
- Verify that the DUT responds to and supports the InitialR2T key during the Login phase. The DUTs response should begin with a capital letter.

- Verify that the DUT responds to and supports the ImmediateData key during the Login phase. The DUTs response should begin with a capital letter.
- Verify that the DUT supports the MaxRecvDataSegmentLength key during the Login phase, no response is expected since this is a declarative key, but the DUT is expected to accept the key.
- Verify that the MaxBurstLength key is responded to properly by the device under test. Verify the value requested falls within the range specified
- Verify that the FirstBurstLength key is transmitted and responded to properly by the device under test. Verify the value requested falls within the value negotiated for MaxBurstLength.
- Verify that a device, which uses the DefaultTime2Retain key, only presents values between 0 - 3600. Verify that a device only uses this key in the leading connection.
- Verify that a device, which uses the DefaultTime2Wait key, only presents values between 0 - 3600. Verify that a device only uses this key in the leading connection.
- Verify that a device, which uses the MaxOutstandingR2T key, only does so in the leading login of a connection, and that the values it presents fall between 1 - 65535.
- Verify that a device, which uses the DataPDUInOrder key, only does so in the leading login of a connection, and that formats the key=value pair properly.
- Verify that a device, which uses the DataSequenceInOrder key, only does so in the leading login of a connection, and that formats the key=value pair properly.
- Verify that a device, which uses the ErrorRecoveryLevel key, only does so in the leading login of a connection, and that formats the key=value pair properly.
- Verify that the Target transmits a Final Login Response (T=1 , NSG=FullFeaturePhase). Verify that this response includes a protocol version and a session ID.
- Verify that a value of '?' is not used during negotiation to indicate 'inquiry'.

Possible Problems: None.

Test #1.2: Standard Login ITT and Version

Purpose: To verify that the DUT properly uses the InitiatorTaskTag, Version Max, and Version Active fields.

Reference: [RFC-7143] Section 4.6.3.2, 11.12.4, 11.12.4.1, 11.13.2

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

Last Modification: October 26, 2015

Discussion:

[RFC-7143] Section 4.6.3.2

> "The Login Phase consists of a sequence of Login Requests and Responses carrying the same Initiator Task Tag."

[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

> "The target MUST use the value presented with the first login request."

[RFC-7143] Section 11.13.2

> "Version-active indicates the highest version supported by the target and initiator. If the target does not support a version within the range specified by the initiator, the target rejects the login and this field indicates the lowest version supported by the target.
>
> All Login responses within the Login Phase MUST carry the same Version-active."

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

Procedure:

- Connect the Testing Station to the Target.
- The Testing Station should attempt to perform an extended login with the target. The Testing Station should attempt to negotiate multiple operational parameters over multiple Login Request PDUs. In all Login Request PDUs transmitted the Testing Station should offer a valid InitiatorTaskTag.
- Any key=value pairs offered by the DUT should be returned in reverse order. For example if the DUT offered FirstBurstLength=1024, MaxBurstLength=2048, the Testing Station should respond with MaxBurstLength=2048, FirstBurstLength=1024,this should not be seen as an error.

Observable Results:

- Verify that at no point in the Login does the device under test alter the InitiatorTaskTag.
- Verify that Version Max field remains the same in all Login Responses, and is the same as presented by the Testing Station.

- Verify that the target offers valid values for version active (0x00).
- Verify that the Status Class and Detail in the responses remains 0x0000.

Possible Problems: None.

Test #2.1: CmdSN

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

Reference: [RFC-7143] Section 11.12.8

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

Last Modification: October 26, 2015

Discussion:
[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 Login Request is a leading Login Request, the target MUST use the value presented in the CmdSN as the target value for the ExpCmdSN."

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a standard login. Use a value of 0 for CmdSN.

Observable Results:

- Verify that a target uses the provided value for CmdSN as the target value for ExpCmdSN.

Possible Problems: None.

Test #3.1: Version Active

Purpose: To verify that the DUT sets the Version Active field properly.

Reference: [RFC-7143] Section 11.12.4, 11.13.2, 11.13.5

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

Last Modification: October 26, 2015

Discussion:
[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 11.13.2

> “Version-active indicates the highest version supported by the target and initiator. If the target does not support a version within the range specified by the initiator, the target rejects the login and this field indicates the lowest version supported by the target.
>
> All Login responses within the Login Phase MUST carry the same Version-active.”

[RFC-7143] Section 11.13.5

> “If the Status-Class [of a Login Response] is not 0, the initiator and target MUST close the TCP connection.”

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a standard login. Transmit a Login Request with a range of Version Max and Min that the target does not support.

Observable Results:

- Verify that the Login Response is a reject and verify that the Version Active field contains the targets lowest supported version, 0x00.
- Verify that the DUT disconnects after sending the Login Reject.

Possible Problems: None.

Test #4.1: T Bit Login Extension

Purpose: To verify that the DUT does attempt to prompt a stage transition, and also that the DUT does not offer parameters for further negotiation while at the same time approving a stage transition prompted by the initiator.

Reference: [RFC-7143] Section 6.3, 11.13.6

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.3

> “Targets MUST NOT submit parameters that require an additional initiator Login Request in a Login Response with the T bit set to 1.”

[RFC-7143] Section 11.13.6

> “A Login Response with the T bit set to 1 MUST NOT contain key=value pairs that may require additional answers from the initiator within the same stage.
>
> If the Status-Class is 0, the T bit MUST NOT be set to 1 if the T bit in the request was 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.
- The Testing Station should transmit a second Login Request with T=1, CSG=Security Negotiation, NSG=Operational Parameter Negotiation. Repeat until the target responds with T=1.
- The Testing Station should now send a Login Request indicating CSG=Operational Parameter Negotiation, T=0. The Testing Station should transmit as many Login PDUs as possible, each offering operational parameters, to expand the time that the devices spend in negotiation.

Observable Results:

- Verify that the Target does not set T=1 to prompt a stage transition, nor offers a value for NSG that exceeds that offered by the Initiator.
- Verify that in the Final Login Response, the target does not include any parameters that require additional negotiation.

Possible Problems: None.

Test #4.2: T Bit NSG Combinations

Purpose: To verify that the DUT uses the T Bit when determining whether the NSG field is reserved or not.

Reference: [RFC-7143] Section 11.12.1, 11.12.3

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

Last Modification: October 27, 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 FullFeaturePhase, then this also indicates that the initiator is ready for the Final Login Response."

[RFC-7143] Section 11.12.3

> "The next stage value [(NSG)] is only valid when the T bit is 1; otherwise, it is reserved."

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 indicate CSG=Security Negotiation, T=0, NSG=2.
- The Testing Station should now set the CSG=Security Negotiation, T=1, NSG=2.

Observable Results:

- Verify that the target does not check the NSG field when T=0, since NSG is reserved when T=0.
- Verify that the target sends a Login Response with Status Class and Status Detail indicating an error was detected with T=1, since NSG is set to 2, is therefore invalid, and not reserved when T=1.

Possible Problems: None.

Test #4.3: T Bit Stage Transition Paths

Purpose: To verify that the DUT is able to make all of the allowable stage transitions and does so when prompted by a Login Request with T=1.

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 27, 2015

Discussion:
[RFC-7143] Section 6.3

> "When a transition is requested by the initiator and acknowledged by the target, both the initiator and target switch to the selected stage."
>
> Only the following Stage transitions are allowed during login: 0-3, 0-1-3, 1-3.

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

Procedure:

- Connect the Testing Station to the DUT.
- Force the device to follow the following stage paths through the login phase 0-3, 0-1-3, 1-3.

Observable Results:

- Verify that the DUT can follow the specified paths.

Possible Problems: None.

Test #4.4: T Bit No Parameters

Purpose: To verify that the DUT will accept a Login Request which contains no parameters for negotiation.

Reference: [RFC-7143] Section 6.3.3

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.3.3

> "Even when the initiator indicates its intent to switch stages by setting the T bit to 1 in a Login Request, the target MAY respond with a Login Response with the T bit set to 0. In that case, the initiator SHOULD continue to set the T bit to 1 in subsequent Login Requests (even empty requests) that it sends, until the target sends a Login Response with the T bit set to 1 or sends a key that requires the initiator to set the T bit 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.
- Once on the OperationalNegotiation stage the Testing Station should send a Login Request PDU with T=0 and no parameters offered for negotiation, wait for a Login Response to be received. The DUT should not treat this an error. Repeat 5 times.
- Transmit a Login Response with T=1, proceed to FullFeaturePhase.

Observable Results:

- Verify that the DUT does not treat the received Request PDUs as errors.

Possible Problems: None.

Test #5.1: ExpStatSN

Purpose: To verify that the DUT ignores the ExpStatSN field when it is reserved.

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 27, 2015

Discussion:
[RFC-7143] Section 11.12.9

> ExpStatSN - “only valid if the Login request restarts a connection,”

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a standard login.
- During the Login set the ExpStatSN field.

Observable Results:

- Verify that the DUT ignores the ExpStatSN field since connection reinstatement is not occuring.

Possible Problems: None.

Test #6.1: Negotiate Once Standard Login

Purpose: To verify that the DUT only transmits a given key=value pair once during the Login Phase negotiations, and that key=value pairs are properly followed by one null character.

Reference: [RFC-7143] Section 6.1, 6.3, 11.13.5

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.1

> "Every key=value pair, including the last or only pair in a LTDS, MUST be followed by one null (0x00) delimiter."

[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). If an attempt to re-negotiate/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error)."

[RFC-7143] Section 11.13.5

> "If the Status Class of a Login Response is not 0, the initiator and target MUST close the TCP connection."

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a standard login.

Observable Results:

- Verify that once a particular parameter negotiation is complete, that it does not appear 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 Boolean Key

Purpose: To verify that the DUT only allows a given key=value pair to appear once during the Login Phase negotiations.

Reference: [RFC-7143] Section 6.3, 11.13.5

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.3

> "If an attempt to re-negotiate/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error)."

[RFC-7143] Section 11.13.5

> "If the Status Class of a Login Response is not 0, the initiator and target MUST close the TCP connection."

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a standard login.
- The Testing Station should offer the Immediate Data parameter twice during the Operational Parameter Negotiation.

Observable Results:

- Verify that the device transmits a Login Reject of reason code initiator error.
- Verify that the DUT disconnects after sending the Login Reject PDU.

Possible Problems: None.

Test #6.3: Negotiate Once Integer Key

Purpose: To verify that the DUT only allows a given key=value pair to appear once during the Login Phase negotiations.

Reference: [RFC-7143] Section 6.3, 11.13.5

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

Last Modification: October 27, 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). If an attempt to re-negotiate/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error)."

[RFC-7143] Section 11.13.5

> "If the Status Class of a Login Response is not 0, the initiator and target MUST close the TCP connection."

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a standard login.
- The Testing Station should offer the MaxBurstLength parameter twice during the Operational Parameter Negotiation.

Observable Results:

- Verify that the device transmits a Login Reject.
- Verify that the DUT disconnects after sending the Login Reject PDU.

Possible Problems: None.

Test #6.4: Negotiate Once List Key

Purpose: To verify that the DUT only allows a given key=value pair to appear once during the Login Phase negotiations.

Reference: [RFC-7143] Section 6.3, 11.13.5

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

Last Modification: October 27, 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). If an attempt to re-negotiate/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error)."

[RFC-7143] Section 11.13.5

> "If the Status Class of a Login Response is not 0, the initiator and target MUST close the TCP connection."

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a standard login.
- The Testing Station should offer the following key=value pair: DataDigest=CHAP, none
- It is expected that the DUT will respond with the key=value pair DataDigest=none, since 'CHAP' would be an invalid value for this key, but 'None' is understood.
- The Testing Station should offer the DataDigest parameter again after the DUT has responded to its initial offer of the DataDigest key. This time the Testing Station should offer DataDigest=CRC32C.

Observable Results:

- Verify that the device transmits a Login Reject.
- Verify that the DUT disconnects after sending the Login Reject PDU.

Possible Problems: None.

Test #6.5: Negotiate Once Same PDU

Purpose: To verify that the DUT only allows a given key=value pair to appear once during the Login Phase negotiations.

Reference: [RFC-7143] Section 6.3, 11.13.5

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

Last Modification: October 27, 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). If an attempt to re-negotiate/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error)."

[RFC-7143] Section 11.13.5

> "If the Status Class of a Login Response is not 0, the initiator and target MUST close the TCP connection."

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a standard login.
- The Testing Station should offer the following key=value pairs in the same PDU: DataDigest=CRC32C, DataDigest=None.

Observable Results:

- Verify that the device transmits a Login Reject.
- Verify that the DUT disconnects after sending the Login Reject PDU.

Possible Problems: None.

Test #7.1: Login Partial Response Standard Request

Purpose: To verify that the DUT constructs a Login Partial Response correctly.

Reference: [RFC-7143] Section 6.3.1

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.3.1

> “The target can answer a received Login Request [with a] . . . Login Response with Login Accept as a partial response (NSG not set to FullFeaturePhase in both request and response) that indicates the start of a negotiation sequence. The response includes the protocol version supported by the target and either security or iSCSI parameters (when no security mechanism is chosen) supported by the target.”

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a login which will prompt the DUT to send a Login Partial Response. Any Login Request with T=0 will do.

Observable Results:

- Verify that the login partial response contains the protocol version supported and either security or operational parameters.

Possible Problems: None.

Test #7.2: Login Partial Response Option Selection

Purpose: To verify that the DUT responds to list negotiations properly with a Login Partial Response.

Reference: [RFC-7143] Section 6.3.2

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.3.2

> "The security exchange sets the security mechanism and authenticates the initiator and the target to each other . . . and is conducted using the key=value parameters carried in the Login Requests and Responses.
>
> An initiator-directed negotiation proceeds as follows:
>
> - The initiator sends a Login Request with an ordered list of the options it supports (authentication algorithm). The options are listed in the initiator's order of preference. The initiator MAY also send private or public extension options.
> - The target MUST reply with the first option in the list it supports and is allowed to use for the specific initiator, unless it does not support any, in which case it MUST answer with "Reject"."

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: DataDigest=CRC32C,Peanutbutter,Jelly,Sandwich,None.

Observable Results:

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

Possible Problems: None.

Test #7.3: Login Partial Response Unsupported Values

Purpose: To verify that the DUT responds to list negotiations properly with a Login Partial Response.

Reference: [RFC-7143] Section 6.3.2

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.3.2

> "The security exchange sets the security mechanism and authenticates the initiator and the target to each other . . . and is conducted using the key=value parameters carried in the Login Requests and Responses.
>
> An initiator-directed negotiation proceeds as follows:
>
> - The initiator sends a Login Request with an ordered list of the options it supports (authentication algorithm). The options are listed in the initiator's order of preference. The initiator MAY also send private or public extension options.
> - The target MUST reply with the first option in the list it supports and is allowed to use for the specific initiator, unless it does not support any, in which case it MUST answer with "Reject"."

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: AuthMethod = SRP.

Observable Results:

- Verify that the device responds with the key=value pair AuthMethod=Reject if none of the offered methods are supported. The device also has the option of transmitting a Login Partial Response with Reject.

Possible Problems: None.

Test #7.4: Login Partial Response Large Values

Purpose: To verify that the DUT handles inadmissible values correctly during Login.

Reference: [RFC-7143] Section 6.2.2

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.2.2

> Simple-Value Negotiations - "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.
- Negotiate a combination of InitialR2T and ImmediateData such that FirstBurstLength is relevant. The Testing Station should transmit a Login Request PDU with the following key=value pair: FirstBurstLength = 16777216. This value is higher than the maximum value specified for FirstBurstLength.

Observable Results:

- Verify that the device responds with a value of Reject, or a number within the valid range for FirstBurstLength. The device also has the option of transmitting a Login Partial Response with Reject.

Possible Problems: If the DUT only supports InitialR2T=Yes ImmediateData=No, the result of this test is "Not Supported".

Test #7.5.1: Login Partial Response ImmediateData

Purpose: To verify that the DUT constructs a Login Partial Response correctly.

Reference: iSCSI Standard Section 6.2.2

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.2.2

> Simple-Value Negotiations - “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.
- The Testing Station should transmit a Login Request PDU with the following key=value pair: ImmediateData = Ok.

Observable Results:

- Verify that the device responds with the key-value pair ImmediateData=Reject, or an admissible value. The device also has the option of transmitting a Login Partial Response with Reject.

Possible Problems: None.

Test #7.5.2: Login Partial Response DataPDUInOrder

Purpose: To verify that the DUT constructs a Login Partial Response correctly.

Reference: [RFC-7143] Section 6.2.2

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.2.2

> Simple-Value Negotiations - “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.
- The Testing Station should transmit a Login Request PDU with the following key=value pair: DataPDUInOrder = Ok.

Observable Results:

- Verify that the device responds with the key-value pair DataPDUInOrder=Reject, or an admissible value. The device also has the option of transmitting a Login Partial Response with Reject.

Possible Problems: None.

Test #7.6: Login Partial Response Invalid Key

Purpose: To verify that the DUT handles inappropriate keys properly during negotiation.

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 27, 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 not understood MUST be key=NotUnderstood."
>
> "The constants "None", "Reject", "Irrelevant", and "NotUnderstood" are reserved and MUST ONLY be used as described here. Violation of this rule is 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.
- The Testing Station should transmit a Login Request PDU with the following key=value pair: ImmediateDate=Yes. Notice 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: Status Detail

Purpose: To verify that the DUT sends a Login Response with appropriate Status Detail codes.

Reference: [RFC-7143] Section 6.3.1, 11.13.5

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 6.3.1

> "The target can answer the [Login Request with]
>
> - Login Response with Login reject. This is an immediate rejection from the target that causes the connection to terminate and the session to terminate if this is the first (or only) connection of a new session."

[RFC-7143] Section 11.13.5

> "If the Status Class is not 0, the initiator and target MUST close the TCP connection."

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 transmits a Login Request with the Version Max field set to 1 and the Version Min field set to 4.

Observable Results:

- Verify that the DUT does not transmit a Login response with a Status Class/ Status Detail of 0x0000, but instead generates a Login Response with an appropriate Status Detail field (0x0205).
- Verify that the DUT disconnects after sending the Login Reject PDU.

Possible Problems: None.

Test #9.1: Invalid PDU During Login

Purpose: To verify that the DUT can identify an Invalid PDU during the Login Phase.

Reference: [RFC-7143] Section 4.2.4, 6.3.1

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

Last Modification: October 27, 2015

Discussion:

[RFC-7143] Section 4.2.4

> “Once the Login Phase has started, if the target receives any PDU except a Login Request, it MUST send a Login reject (with Status "invalid during login") and then disconnect.”

[RFC-7143] Section 6.3.1

> Login Response with Login reject – “The T bit, the CSG field, and the NSG field are reserved”

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

Procedure:

- Connect the Testing Station to the DUT.
- Transmit a valid Login Request. Wait for a Login Response.
- Transmit a SCSI Command PDU to the device.

Observable Results:

- Verify that the DUT does not transmit a Login response with a Status Class/ Status Detail of 0x0000, but instead generates a Login Response with an appropriate Status Detail field (0x020B).
- Verify that the T bit, NSG, and CSG are all set to 0.
- Verify that the DUT disconnects after sending the Login Reject PDU.

Possible Problems: None.

Test #9.2: Invalid PDU Before Login

Purpose: To verify that the DUT can identify an Invalid PDU before the Login Phase begins.

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 27, 2015

Discussion:
[RFC-7143] Section 4.2.4

> "A target receiving any PDU except a Login Request before the Login phase is started MUST immediately terminate the connection on which the PDU was received."

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

Procedure:

- Connect the Testing Station to the DUT.
- Do not transmit an initial valid Login Request; instead transmit a SCSI Command PDU to the device.

Observable Results:

- Verify that the DUT does not transmit any Login response but instead terminates the connection immediately.

Possible Problems: None.

Test #10.1: Parameter Names

Purpose: To verify that the DUT properly formats all key=value pairs.

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 27, 2015

Discussion:
[RFC-7143] Section 6.1

> "Every key=value pair, including the last or only pair in a LTDS, MUST be followed by one null (0x00) delimiter.
>
> A key-name is whatever precedes the first = in the key=value pair. The term "key" is used frequently in this document in place of "key-name".
>
> A value is whatever follows the first "=" in the key=value pair up to the end of the key=value pair, but not including the null delimiter."

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a Standard Login. Allow the DUT to transmit parameters for negotiation.

Observable Results:

- Verify that all parameter names appear in the format key=value format described above.
- Verify that the key=value pair is followed by one null (0x00) delimiter.
- Verify that a value of '?' does not appear in any negotiations, and that the first letter for all keys and values are capitalized.

Possible Problems: None.

Test #11.1: AuthMethod

Purpose: To verify that if the DUT supports any AuthMethod, it supports 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 27, 2015

Discussion:

[RFC-7143] Section 12.1

> “The initiator and target MUST implement CHAP. All other authentication methods are OPTIONAL.
>
> Private or public extension algorithms MAY also be negotiated for authentication methods. Whenever a private or public extension algorithm is part of the default offer . . . the implementer MUST ensure that CHAP is listed as an alternative in the default offer, and “None” is not part of the default offer.”

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a Standard Login.
- In the Security Negotiation stage, offer a Login Request PDU without any AuthMethod parameters. This will give the DUT the opportunity to offer AuthMethod keys.
- If the DUT does not offer any AuthMethod keys, transmit a Login Request PDU with the following: AuthMethod= CHAP, SRP, KRB5, SPKM1, SPKM2, None.

Observable Results:

- If the device offers an AuthMethod, that CHAP is included in the list and that SPKM1 and SPKM2 are not in the list.
- If the device does not offer an 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).

Possible Problems: None.

Test #12.1: Header and Data Digest Default Values

Purpose: To verify that the DUT properly negotiates values for 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 27, 2015

Discussion:
[RFC-7143] Section 13.1

> “The following table lists cyclic integrity checksums that can be negotiated for the digests and MUST be implemented by every iSCSI initiator and target. These digest options only have error detection significance.
>
> | Name | Description | Generator |
> |---|---|---|
> | CRC32C | 32-bit CRC | 0x11edc6f41 |
> | None | no digest | |
>
> ”
>
> “Support for public or private extension digests is OPTIONAL.”

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a Standard Login.

Observable Results:

- Verify if the device attempts a Header or Data Digest negotiation, it offers CRC32C or none as options.

Possible Problems: None.

Test #12.2: Header and Data Digest Proprietary Values

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 13.1

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 13.1

> “The following table lists cyclic integrity checksums that can be negotiated for the digests and MUST be implemented by every iSCSI initiator and target. These digest options only have error detection significance.

```
+---------------------------------------------+
| Name          | Description     | Generator |
+---------------------------------------------+
| CRC32C        | 32-bit CRC      |0x11edc6f41|
+---------------------------------------------+
| None          | no digest                   |
+---------------------------------------------+”
```

> “Support for public or private extension digests is OPTIONAL.”

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a Standard Login.
- Offer the following key=value pairs: HeaderDigest=Y-edu.unh.testor, None; DataDigest=Y-edu.unh.testor, None

Observable Results:

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

Possible Problems: None.

Test #12.3: Header and Data Digest Support CRC32C

Purpose: To verify that the DUT supports CRC32C for both Header Digest 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 27, 2015

Discussion:
[RFC-7143] Section 13.1

> “The following table lists cyclic integrity checksums that can be negotiated for the digests and MUST be implemented by every iSCSI initiator and target. These digest options only have error detection significance.

```
+---------------------------------------------+
| Name          | Description     | Generator |
+---------------------------------------------+
| CRC32C        | 32-bit CRC      |0x11edc6f41|
+---------------------------------------------+
| None          | no digest                   |
+---------------------------------------------+”
```

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a Standard Login.
- Offer the following key=value pairs: HeaderDigest=CRC32C; DataDigest=CRC32C

Observable Results:

- Verify that the device responds with the value 'CRC32C' for both Header and Data Digest.

Possible Problems: None.

Test #13.1: MaxConnections

Purpose: To verify that the DUT properly negotiates a value for 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 27, 2015

Discussion:
[RFC-7143] Section 13.2

> MaxConnections is defined for use only on the leading connection of a session and is a numerical value from 1 to 65535.

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

Procedure:

- Connect the Testing Station to the DUT.
- Perform a Standard Login, offer the key MaxConnections=65535.

Observable Results:

- Verify that if the DUT attempts to negotiate MaxConnections, it only does so in the leading login of a connection.
- Verify that the desired MaxConnections value falls within the required range of 1 to 65535.

Possible Problems: None.

Test #14.1: TargetAlias

Purpose: To verify that the DUT properly offers a value for TargetAlias.

Reference: [RFC-7143] Section 13.6

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

Last Modification: October 27, 2015

Discussion:

[RFC-7143] Section 13.6

> "If a target has been configured with a human-readable name or description, this name SHOULD be communicated to the initiator during a Login Response PDU if SessionType=Normal (see Section 13.21). This string is not used as an identifier, nor is it meant to be used for authentication or authorization decisions. It can be displayed by the initiator's user interface in a list of targets to which it is connected."

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

Procedure:

- Configure the DUT with a human readable name.
- Connect the Testing Station to the DUT and perform a standard login.

Observable Results:

- Verify that the target communicates the TargetAlias.

Possible Problems: If the DUT cannot be configured with a TargetAlias, the result of this test is "Not Supported".

Test #15.1: 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 27, 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:

- Connect the Testing Station to the DUT and perform a standard login.
- Transmit a request with the OFMarker=Yes key to the Device Under Test.
- Transmit a request with the IFMarker=Yes key to the Device Under Test.
- Transmit a request with the OFMarkInt range of 1 to 65535.
- Transmit a request with the IFMarkInt range of 1 to 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 #16.1: FirstBurstLength Exceeds MaxBurstLength

Purpose: To verify that the DUT properly negotiates values for FirstBurstLength and MaxBurstLength.

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 27, 2015

Discussion:
[RFC-7143] Section 13.14

> FirstBurstLength – "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 SCSI command. This covers the immediate data (if any) and the sequence of unsolicited Data-Out PDUs (if any) that follow the command.
>
> FirstBurstLength must not exceed MaxBurstLength."
>
> FirstBurstLength is defined for use only in the leading login of a session. It is a numerical value from 512 to 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 and perform a standard login.
- Negotiate a combination of InitialR2T and ImmediateData so that FirstBurstLength is relevant.
- Offer the MaxBurstLength key.
- In the second Login Request transmit a request with the FirstBurstLength key, greater than the MaxBurstLength key.

Observable Results:

- Verify that the FirstBurstLength key is either rejected by the DUT, or the DUT offers a value for FirstBurstLength that falls within the legal range.

Possible Problems: None.

Test #16.2: FirstBurstLength Within MaxBurstLength

Purpose: To verify that the DUT properly negotiates values for FirstBurstLength and MaxBurstLength.

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 27, 2015

Discussion:
[RFC-7143] Section 13.14

> FirstBurstLength – "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 SCSI command. This covers the immediate data (if any) and the sequence of unsolicited Data-Out PDUs (if any) that follow the command.
>
> FirstBurstLength must not exceed MaxBurstLength."
>
> FirstBurstLength is defined for use only in the leading login of a session. It is a numerical value from 512 to 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 and perform a standard login.

Observable Results:

- If the device offers the FirstBurstLength key, verify that it is not greater than the MaxBurstLength key.

Possible Problems: If FirstBurstLength is irrelevant due to values negotiated for ImmediateData and InitialR2T, the result of this test is "Not Supported".

Test #16.3: FirstBurstLength Default Exceeds MaxBurstLength

Purpose: To verify that the DUT properly negotiates values for FirstBurstLength and MaxBurstLength. Intended to be informative only.

Reference: [RFC-7143] Section 13.14, 11.13.5

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

Last Modification: October 27, 2015

Discussion:
[RFC-7143] Section 13.14

> FirstBurstLength – "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 SCSI command. This covers the immediate data (if any) and the sequence of unsolicited Data-Out PDUs (if any) that follow the command.
>
> FirstBurstLength must not exceed MaxBurstLength."
>
> FirstBurstLength is defined for use only in the leading login of a session. It is a numerical value from 512 to 2^{24}-1.

[RFC-7143] Section 11.13.5

> "If the Status Class of a Login Response is not 0, the initiator and target MUST close the TCP connection."

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 perform a standard login.
- Negotiate InitialR2T and ImmediateData so that FirstBurstLength is relevant.
- The Testing Station offers a MaxBurstLength key which is less than the default value of FirstBurstLength.

Observable Results:

- If the device offers the FirstBurstLength key, verify that it is not greater than the MaxBurstLength key.
- The DUT should attempt to negotiate a value for FirstBurstLength, which is smaller than the value negotiated for MaxBurstLength. Another option for the DUT is to transmit a Login Reject PDU and disconnect, or reject the value offered by the Testing Station for MaxBurstLength.

Possible Problems: If FirstBurstLength is irrelevant due to values negotiated for ImmediateData and InitialR2T, the result of this test is "Not Supported".

Test #16.4: FirstBurstLength Exceeds Default MaxBurstLength

Purpose: To verify that the DUT properly negotiates values for FirstBurstLength and MaxBurstLength. Intended to be informative only.

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 27, 2015

Discussion:
[RFC-7143] Section 13.14

> FirstBurstLength – "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 SCSI command. This covers the immediate data (if any) and the sequence of unsolicited Data-Out PDUs (if any) that follow the command.
>
> FirstBurstLength must not exceed MaxBurstLength."
>
> FirstBurstLength is defined for use only in the leading login of a session. It is a numerical value from 512 to 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 and perform a standard login.
- Negotiate InitiatlR2T and ImmediateData such that FirstBurstLength is relevant.
- The Testing Station offers a FirstBurstLength key of size greater than the default value for MaxBurstLength, but still a legal value.

Observable Results:

- The DUT should offer a value for FirstBurstLength that is smaller than the default value for MaxBurstLength. Alternatively the DUT could attempt to negotiate a value for MaxBurstLength which was greater than the value negotiated for FirstBurstLength.

Possible Problems: If FirstBurstLength is irrelevant due to values negotiated for ImmediateData and InitialR2T, the result of this test is "Not Supported".

Test #17.1: SessionType

Purpose: To verify that the DUT properly handles the SessionType key.

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 27, 2015

Discussion:
[RFC-7143] Section 13.21

> SessionType – "The initiator indicates the type of session it wants to create. The target can accept or reject it."
>
> SessionType is defined only for use in the Leading Login of a connection; it may only be sent by the Initiator. Defined values for this key are "Discovery" and "Normal".
>
> "The Discovery session implies MaxConnections = 1 and overrides both the default and an explicit setting."

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 perform a standard login.
- Transmit a request with the key=value pair SessionType=Discovery.

Observable Results:

- Verify that the DUT accepts or rejects the key=value pair SessionType=Discovery.

Possible Problems: None.

Test #18.1: C Bit

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

Reference: [RFC-7143] Section 6.1, 6.2, 11.12.2

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

Last Modification: October 27, 2015

Discussion:

[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."

[RFC-7143] Section 11.12.2

> C (Continue) bit - "When set to 1, this bit indicates that the text (set of key=value pairs) in this Login Request is not complete (it will be continued on subsequent Login Requests); otherwise, it indicates that this Login Request ends a set of key=value pairs."

[RFC-7143] Section 6.2

> "A target receiving a Text or Login Request with the C bit set to 1 MUST answer with a Text or Login Response with no data segment (DataSegmentLength 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' of 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.
- Transmit a READ command to the DUT. Wait for Data-in PDUs.

Observable Results:

- The DUT should transmit 'NotUnderstood' to the vendor specific keys. The DUT should not disconnect.
- Verify that the DUT responds the received Login Request which had the C bit set to 1, with a Login Response with no data segment.
- Verify that the DUT adheres to the MaxRecvDataSegmentLength declared by the Testing Station when sending Data-Out PDUs.

Possible Problems: None.

Test #19.1: Errors Invalid Keys

Purpose: To verify that the DUT recognizes keys that are invalid for an initiator 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 Request with the following key=value pairs: TargetAlias=UNHIOL, TargetPortalGroupTag=1, TargetAddress=10.0.0.1:3260,1.

Observable Results:

- Verify that the DUT transmits a Login Partial Response with Reject to indicate that it has detected the error.
- The DUT should not transmit key=Reject or key=NotUnderstood to these keys. While this behavior would not technically violate the standard, a robust implementation should make use of the Reject functionality.
- Alternatively, the DUT may choose to simply disconnect or to ignore this error.

Possible Problems: None.

Test #19.2.1: Errors X Keys NotUnderstood

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 27, 2015

Discussion:
[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."
>
> "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."
>
> If an iSCSI device does not recognize a vendor specific X key, it should reply with the value "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 Request with the following key=value pair: X-edu.unh.iol-extension-key-1=test

Observable Results:

- Verify that the DUT replies with the value 'NotUnderstood'.

Possible Problems: None.

Test #19.2.2: Errors X Keys Too Long (Informative)

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 27, 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 Request 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 responds with a Login Response with Status Class = 0x02 (Initiator Error) and/or 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: This is an informative test, as iSCSI Initiators and Targets are not required to do exhaustive protocol conformance checking on incoming iSCSI PDUs.

Test #19.3.1: Errors Big Values Simple 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 27, 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 Request with the following keys=value pair:
  ImmediateData=NoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoN
  oNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNo
  NoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNo
  NoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNo
  NoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNo.

Observable Results:

- The DUT should reject the received value. The DUT may also choose to send Login Reject and terminate the connection. The DUT may also choose to reply with an admissible value for the given key.

Possible Problems: None.

Test #19.3.2: Errors Big Values Declared Value (Informative)

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

Reference: [RFC-7143] Section 6.1, 7.10, 13.7

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

Last Modification: October 27, 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 13.7

> The InitiatorAlias 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 Request with the following keys=value pair: InitiatorAlias = SuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiator

Observable Results:

- The DUT may send Login Reject and terminate the connection.

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 an iSCSI device may only read 255 bytes of data since that is all that is valid. Therefore the device 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 #19.4: 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 27, 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 Response with the following key=value pair: MaxConnections=?

Observable Results:

- The DUT should reject the received key. The DUT may also choose to send Login Reject and terminate the connection. The DUT may also choose to reply with an admissible value for the given key.

Possible Problems: None.

Test #20.1: TargetPortalGroupTag Normal

Purpose: To see if the DUT properly includes the TargetPortalGroupTag key during the Login Phase.

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 27, 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:

- Transmit a Login Request with key=value pair SessionType=Normal to the DUT.

Observable Results:

- Verify that the DUT transmits a Login Response with the TargetPortalGroupTag key included, with a valid value.

Possible Problems: None.

Test #21.1: Irrelevant Keys

Purpose: To see if 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 27, 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:

- Transmit a Login Request with key=value pair SessionType=Discovery to the DUT.
- After receiving a Login Response, transmit a Login Request with the following keys, each with a valid value: MaxConnections=10, InitialR2T=No, ImmediateData=Yes, MaxBurstLength=2**24-1, FirstBurstLength=2**24-1, MaxOutstandingR2T=10, DataPDUInOrder=No, DataSequenceInOrder=No, TaskReporting=RFC3720.

Observable Results:

- Verify that the DUT transmits a Login Response with good status. Verify that the DUT transmitted either Irrelevant or a valid value for each key offered.

Possible Problems: None.

Test #22.1: 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 27, 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:

- Transmit a Login Request with key=value pairs SessionType=Discovery and ErrorRecoveryLevel=1 to the DUT.

Observable Results:

- The DUT should negotiate ErrorRecoveryLevel=0.

Possible Problems: None.

Test #23.1: NotUnderstood for Required Keys

Purpose: To verify that the DUT treats a response of NotUnderstood as a protocol error for keys defined in
RFC3720.

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 27, 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:

- Transmit a Login Request with key=value pair SessionType=Normal to the DUT.
- After receiving the first Login Response from the DUT, which should contain the TargetPortalGroupTag key, transmit a Login Request with the key=value pair TargetPortalGroupTag=NotUnderstood to the DUT.

Observable Results:

- Verify that the DUT responds to the second PDU by transmitting a Login Response with an appropriate response code indicating initiator error.
- Verify that the DUT disconnects due to a protocol error.

Possible Problems: None.

Test #24.1: 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 27, 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:

- Transmit a Login Request with key=value pairs SessionType=Normal and TaskReporting=RFC3720,ResponseFence,FastAbort to the DUT.

Observable Results:

- Verify that the DUT chooses a value for the TaskReporting key from the list offered by the DUT.

Possible Problems: None.

Test #25.1: iSCSIProtocolLevel (Informative)

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

Reference: [RFC-7143] Section 13.24, 7.1.1

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

Last Modification: October 27, 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:

- Transmit a Login Request with key=value pairs SessionType=Normal and iSCSIProtocolLevel=1.

Observable Results:

- Verify that the DUT 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 #26.1 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 29, 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 #27.1 Receive Limit During Login

Purpose: To verify that the DUT properly handles a Login Request 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 29, 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 Request 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.
- Transmit a READ command for 2048 byets.

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.