

UNH IOL iSCSI CONSORTIUM

Login Phase Test Suite for iSCSI Initiators *Version 1.2*

Technical Document



Last Updated January 4, 2007

© 2006 University of New Hampshire InterOperability Laboratory

*UNH-IOL iSCSI Consortium
InterOperability Laboratory
University of New Hampshire*

*121 Technology Drive, Suite 2
Durham, NH 03824
Phone: (603) 862-1908
Fax: (603) 862-4181*

<http://www.iol.unh.edu/consortiums/iscsi>

The University of New Hampshire
InterOperability Laboratory

TABLE OF CONTENTS

MODIFICATION RECORD4

ACKNOWLEDGMENTS5

INTRODUCTION.....6

REFERENCES.....8

TEST SETUP.....9

GROUP 1: LOGIN PHASE FOR INITIATORS10

TEST #1.1 STANDARD LOGIN 11

TEST #2.1 TSIH 12

TEST #2.2 TSIH 13

TEST #2.3 TSIH 14

TEST #3.1 T-BIT 15

TEST #3.2 T-BIT 16

TEST #4.1 STATSN..... 17

TEST #4.2 STATSN..... 18

TEST #5.1 EXPSTATSN 19

TEST #6.1 NEGOTIATE ONCE..... 20

TEST #6.2 NEGOTIATE ONCE..... 21

TEST #6.3 NEGOTIATE ONCE..... 22

TEST #7.1 LOGIN REQUEST..... 23

TEST #7.2: LOGIN REQUEST 24

TEST #7.3: LOGIN REQUEST 25

TEST #7.4: LOGIN REQUEST 26

TEST #7.5: LOGIN REQUEST 27

TEST #7.6: LOGIN REQUEST 28

TEST #8.1 INVALID PDU 29

TEST #9.1 HEADER AND DATA DIGESTS 30

TEST #9.2: HEADER AND DATA DIGEST 31

TEST #10.1 MAXCONNECTIONS..... 32

TEST #11.1 INITIATOR NAME TARGET NAME 33

TEST #12.1 MARKER NEGOTIATION..... 34

TEST #13.1 BOOLEAN NEGOTIATION..... 35

TEST #13.2 LEGAL BOOLEAN NEGOTIATION 36

TEST #13.3 ILLEGAL BOOLEAN NEGOTIATION 37

TEST #14.1 MAXRECVDATASEGMENTLENGTH 38

TEST #15.1 MAXBURSTLENGTH 39

TEST #16.1 FIRSTBURSTLENGTH 40

TEST #16.2 FIRSTBURSTLENGTH 41

TEST #16.3 FIRSTBURSTLENGTH 42

TEST #17.1 DEFAULTTIME2RETAIN 43

*The University of New Hampshire
InterOperability Laboratory*

TEST #18.1 DEFAULTTIME2WAIT	44
TEST #19.1 MAXOUTSTANDINGR2T	45
TEST #20.1 ERRORRECOVERYLEVEL	46
TEST #21.1 SESSIONTYPE	47
TEST #22.1 AUTHMETHOD	48
TEST #23.1 TARGETPORTALGROUPTAG	49
TEST #24.1 C BIT	50
TEST #25.1 REDIRECT	51
TEST #26.1 ERRORS INVALID KEYS	52
TEST #26.2.1 ERRORS X KEYS	53
TEST #26.2.2 ERRORS X KEYS	54
TEST #26.3.1 ERRORS BIG VALUES	55
TEST #26.3.2 ERRORS BIG VALUES	56
TEST #26.4 ERRORS INQUIRE VALUE	57
TEST #27.1 IRRELEVANT KEYS	58
APPENDICES	59

*The University of New Hampshire
InterOperability Laboratory*

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.

The University of New Hampshire
InterOperability Laboratory

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

The University of New Hampshire
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 3720 iSCSI* (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 number (e.g., [1]) 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 University of New Hampshire
InterOperability Laboratory

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

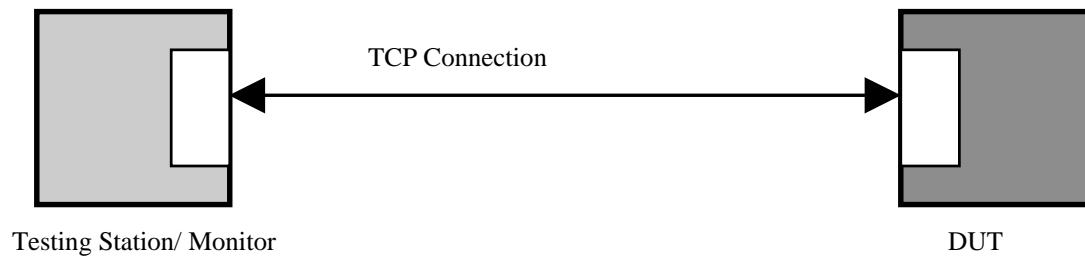
The following documents are referenced in this text:

IETF IPS Working Group iSCSI RFC 3720

TEST SETUP

The following test setup is used in this test suite:

Test Setup 1:



GROUP 1: LOGIN PHASE FOR INITIATORS

Overview: This group of tests verifies the Login Phase specifications of iSCSI defined in RFC 3720. Comments and questions regarding the implementation of these tests are welcome, and may be forwarded to Peter Scruton at the UNH InterOperability Lab (pjs@iol.unh.edu).

*The University of New Hampshire
InterOperability Laboratory*

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: iSCSI Standard 3.5.3.2; 10.12.7; 10.12.4; 10.12.8; 10.12.5

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

Last Modification: February 1, 2005

Discussion: 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. 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.

A Login Requests within the Login Phase MUST carry the same Version-max and the same Version-min. The only valid Version Number is 0x00 as of RFC 3720. Login Responses must contain the same Version Max and Version Min values as the initial Login request.

CmdSN is the initial command sequence number of a session. The target MUST use the value for CmdSN presented with the first login request. If the leading login carries the CmdSN 123, all other login requests in the same login phase carry the CmdSN 123, and this first non-immediate command in Full Feature Phase also carries the CmdSN 123.

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

*The University of New Hampshire
InterOperability Laboratory*

Test #2.1 TSIH

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

Reference: iSCSI Standard 10.12.6

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

Last Modification: February 1, 2005

Discussion: For a new session, the first Login Request must have a TSIH of zero. Otherwise, the TSIH sent by the target at the conclusion of the successful login of the first connection for this session **MUST** be used. 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 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

*The University of New Hampshire
InterOperability Laboratory*

Test #2.2 TSIH

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

Reference: iSCSI Standard 10.12.6, 10.13.3

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

Last Modification: February 1, 2005

Discussion: The reserved value 0 MUST be used for TSIH 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. With the exception of the Login Final-Response in a new session, the TSIH field should be set to the TSIH provided by the initiator in the Login Request.

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

*The University of New Hampshire
InterOperability Laboratory*

Test #2.3 TSIH

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

Reference: iSCSI Standard 10.12.6

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

Last Modification: February 1, 2005

Discussion: For a new session, the TSIH is zero. For a connection added to an existing session the TSIH sent by the target at the conclusion of the successful login of the first connection for the 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: None

*The University of New Hampshire
InterOperability Laboratory*

Test #3.1 T-Bit

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

Reference: iSCSI Standard 10.12.1

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

Last Modification: February 1, 2005

Discussion: If the T bit in the Login Request is set to 1, it 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

*The University of New Hampshire
InterOperability Laboratory*

Test #3.2 T-Bit

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

Reference: iSCSI Standard 10.12.1

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

Last Modification: February 1, 2005

Discussion: If the T bit is set to 1, 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 also 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

*The University of New Hampshire
InterOperability Laboratory*

Test #4.1 StatSN

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

Reference: iSCSI Standard 10.13.4

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

Last Modification: February 1, 2005

Discussion: The StatSN field is assigned and incremented by the target. For the first Login Response 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 the previous StatSN + 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

*The University of New Hampshire
InterOperability Laboratory*

Test #4.2 StatSN

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

Reference: iSCSI Standard 10.13.4

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

Last Modification: Mon May 19 15:48:35 2003

Discussion: StatSN is assigned and incremented by the target. 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

*The University of New Hampshire
InterOperability Laboratory*

Test #5.1 ExpStatSN

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

Reference: iSCSI Standard 10.12.9

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

Last Modification: Mon May 19 15:48:48 2003

Discussion: 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.

The University of New Hampshire
InterOperability Laboratory

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: iSCSI Standard 5.3

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

Last Modification: Mon May 19 15:48:57 2003

Discussion: 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/re-declare parameters not specifically allowed is 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.

*The University of New Hampshire
InterOperability Laboratory*

Test #6.2 Negotiate Once

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

Reference: iSCSI Standard 5.3

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

Last Modification: Mon May 19 15:49:14 2003

Discussion: 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/re-declare parameters not specifically allowed is 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 4.3. 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

*The University of New Hampshire
InterOperability Laboratory*

Test #6.3 Negotiate Once

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

Reference: iSCSI Standard 5.3

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

Last Modification: Mon May 19 15:52:07 2003

Discussion: 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/re-declare parameters not specifically allowed is 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 4.3. 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

*The University of New Hampshire
InterOperability Laboratory*

Test #7.1 Login Request

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

Reference: iSCSI Standard 5.3

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

Last Modification: February 1, 2005

Discussion: The initial login request of any connection **MUST** include the following key=value pairs: InitiatorName, Target Name. The Login Phase **MAY** include a Security Negotiation stage and an Operational Negotiation 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 Security Negotiation **MUST** precede the Login Operational Negotiation.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #7.2: Login Request

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

Reference: iSCSI Standard 5.2.1, 12.1

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

Last Modification: February 1, 2005

Discussion: 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.

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, the following key=value pair: 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 DataDigest in the first Login Request, a different key=value pair with list parameters may be offered by the Testing Station in the first Login Response.

*The University of New Hampshire
InterOperability Laboratory*

Test #7.3: Login Request

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

Reference: iSCSI Standard 5.2.1, 11.1

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

Last Modification: February 1, 2005

Discussion: 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 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.

*The University of New Hampshire
InterOperability Laboratory*

Test #7.4: Login Request

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

Reference: iSCSI Standard 5.2.2, 12.14

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

Last Modification: Mon May 19 15:57:31 2003

Discussion: In numerical value negotiation, the originator sends a value for a given key=value pair. The responding party **MUST** respond with the same key and a value that it supports and is allowed to use for the specific key.

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: This item is Not Testable if FirstBurstLength is Irrelevant.

*The University of New Hampshire
InterOperability Laboratory*

Test #7.5: Login Request

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

Reference: iSCSI Standard 5.2.2, 12.11

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

Last Modification: Mon May 19 15:57:42 2003

Discussion: In Boolean value negotiation, the originator sends a value for a specific key. The responding party MUST respond with the same key and a value that it supports (and is allowed to use for the specific originator).

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:

- Verify that the device responds with the key-value pair Immediate Data=Reject or selects an admissible value.

Possible Problems: None.

*The University of New Hampshire
InterOperability Laboratory*

Test #7.6: Login Request

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

Reference: iSCSI Standard 5.2

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

Last Modification: Mon May 19 15:57:52 2003

Discussion: 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.

*The University of New Hampshire
InterOperability Laboratory*

Test #8.1 Invalid PDU

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

Reference: iSCSI Standard 3.2.3

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

Last Modification: Mon May 19 15:58:08 2003

Discussion: 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.

*The University of New Hampshire
InterOperability Laboratory*

Test #9.1 Header and Data Digests

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

Reference: iSCSI Standard 12.1

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

Last Modification: February 11, 2005

Discussion: CRC32C and None must be offered as options for either Header or Data Digest. Proprietary algorithms may be listed.

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 'None' as options.

Possible Problems: None.

*The University of New Hampshire
InterOperability Laboratory*

Test #9.2: Header and Data Digest

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: iSCSI Standard 12.1

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

Last Modification: February 11, 2005

Discussion: CRC32C and None are the only options that a device must offer for either Header or Data Digest. Proprietary algorithms may be listed. Even if the response will be 'None' the DUT must transmit a response. In this test the Testing Station will offer an unknown value and None as values for the HeaderDigest key. The DUT is expected to respond with a value of None. The DUT must support this value. The list negotiation is only considered failed if no values are offered that the DUT supports.

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: None.

*The University of New Hampshire
InterOperability Laboratory*

Test #10.1 MaxConnections

Purpose: To verify that the DUT properly negotiates MaxConnections.

Reference: iSCSI Standard 12.2

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

Last Modification: Mon May 19 15:58:28 2003

Discussion: MaxConnections can only be negotiated in the leading connection of a session. MaxConnections can range from 1 - 65535. No 'zero' or 'don't care' value is allowed.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #11.1 Initiator Name Target Name

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

Reference: iSCSI Standard 5.3, 12.4, 12.5

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

Last Modification: Mon May 19 15:58:39 2003

Discussion: The initiator of the TCP connection MUST provide the TargetName 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. The initiator of the TCP connection MUST provide the InitiatorName 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. InitiatorName and TargetName MUST not be redeclared within the login phase.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #12.1 Marker Negotiation

Purpose: To verify that the DUT properly negotiates OFMarker, IFMarker, OFMarkInt, IFMarkInt.

Reference: iSCSI Standard 5.2, A.3.1, A.3.2

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

Last Modification: February 11, 2005

Discussion: OFMarker is used to turn on or off the initiator to target markers on the connection. IFMarker is used to turn on or off the target to initiator markers on the connection. The Default for each is No. OFMarkInt is used to set the interval for the initiator to target markers on the connection. IFMarkInt is used to set the interval for the target to initiator markers on the connection. The Default is 2048. OFMarkInt is Irrelevant when: OFMarker=No. IFMarkInt is Irrelevant when: IFMarker=No. If a specific key is not relevant for the current negotiation, the acceptor may answer with the constant "Irrelevant" for all types of negotiation. However the negotiation is not considered as failed if the answer is "Irrelevant". The "Irrelevant" answer is meant for those cases in which several keys are presented by a proposing party but the selection made by the acceptor for one of the keys makes other keys irrelevant.

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: OFMarker=Yes, IFMarker=Yes, OFMarkInt range of 1~65535, IFMarkInt range of 1~65535.

Observable Results:

- Verify that the response to both OFMarker and IFMarker is Yes or No.
- Verify that the device responds with a value within the specified range for IFMarkInt, OFMarkInt, or 'Irrelevant' of the response to OFMarker and IFMarker was 'No'.

Possible Problems: The DUT should not transmit OFMarkInt=Reject or IFMarkInt=Reject, since the range offered encompassed all legal values. In this case the Reject value is intended for the case where OFMarker and/or IFMarker were already negotiated to 'Yes'.

*The University of New Hampshire
InterOperability Laboratory*

Test #13.1 Boolean Negotiation

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

Reference: iSCSI Standard 5.2.2, 12.10, 12.11, 12.18, 12.19

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

Last Modification: February 11, 2005

Discussion: 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. InitialR2T, DataSequenceInOrder, DataPDUInOrder each have the result function OR. Immediate Data has the result function AND.

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.

*The University of New Hampshire
InterOperability Laboratory*

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: iSCSI Standard 5.2.2, 12.10, 12.11, 12.18, 12.19

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

Last Modification: Thu Jun 19 16:15:54 2003

Discussion: For Boolean negotiations 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. InitialR2T, DataSequenceInOrder, DataPDUInOrder each have the result function OR. Immediate Data has the result function AND.

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 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: None.

The University of New Hampshire
InterOperability Laboratory

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: iSCSI Standard 5.2.2, 12.10, 12.11, 12.18, 12.19

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

Last Modification: Tue Jun 3 13:55:24 2003

Discussion: 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. InitialR2T, DataSequenceInOrder, DataPDUInOrder each have the result function OR. Immediate Data has the result function AND.

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 this item is not testable.

*The University of New Hampshire
InterOperability Laboratory*

Test #14.1 MaxRecvDataSegmentLength

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

Reference: iSCSI Standard 12.12

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

Last Modification: Mon May 19 15:59:45 2003

Discussion: With the MaxRecvDataSegmentLength key 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. MaxRecvDataSegmentLength can be a numerical-value-512-to-(2**24-1) the default is 8192 bytes.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #15.1 MaxBurstLength

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

Reference: iSCSI Standard 12.13

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

Last Modification: Mon Jul 28 08:29:34 2003

Discussion: MaxBurstLength key can only be used in the Leading Login of a session. The MaxBurstLength key is used to declare the maximum SCSI data payload in bytes in a Data-In or a solicited Data-Out iSCSI sequence and must be between 512 and $2^{24}-1$. Default is 262144 A 'don't care' zero value is not allowed.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #16.1 FirstBurstLength

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

Reference: iSCSI Standard 12.14

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

Last Modification: Mon May 19 16:00:00 2003

Discussion: The FirstBurstLength key can only be used in the leading login of a session. The FirstBurstLength key is used to 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 range for FirstBurstLength is 512 to $2^{24}-1$ bytes. A 'don't care' zero value is not allowed.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #16.2 FirstBurstLength

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

Reference: iSCSI Standard 12.14

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

Last Modification: Mon May 19 16:00:16 2003

Discussion: The FirstBurstLength key can only be used in the leading login of a session. The FirstBurstLength key is used to 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 range for FirstBurstLength is 512 to $2^{24}-1$ bytes. A 'don't care' zero value is not allowed.

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, this item is not testable.

*The University of New Hampshire
InterOperability Laboratory*

Test #16.3 FirstBurstLength

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

Reference: iSCSI Standard 12.14

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

Last Modification: Mon May 19 16:00:21 2003

Discussion: The FirstBurstLength key can only be used in the leading login of a session. The FirstBurstLength key is used to 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 range for FirstBurstLength is 512 to $2^{24}-1$ bytes. A 'don't care' zero value is not allowed.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #17.1 DefaultTime2Retain

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

Reference: iSCSI Standard 12.16

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

Last Modification: February 23, 2005

Discussion: The DefaultTime2Retain can only be used in the leading connection of a session and must be an integer from 0 - 3600.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #18.1 DefaultTime2Wait

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

Reference: iSCSI Standard 12.15

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

Last Modification: February 23, 2005

Discussion: The DefaultTime2Wait can only be used in the leading connection and must be an integer from 0 - 3600.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #19.1 MaxOutstandingR2T

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

Reference: iSCSI Standard 12.17

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

Last Modification: February 23, 2005

Discussion: The MaxOutstandingR2T key can only be used in the leading login of a session and must be a number from 1 - 65535. A 'don't care' zero value is not allowed.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #20.1 ErrorRecoveryLevel

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

Reference: iSCSI Standard 12.20

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

Last Modification: Mon May 19 16:01:21 2003

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

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:

- Verify that a device responds to the ErrorRecoveryLevel key, with 0, 1, or 2.

Possible Problems: None.

*The University of New Hampshire
InterOperability Laboratory*

Test #21.1 SessionType

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

Reference: iSCSI Standard 12.21

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

Last Modification: Mon May 19 16:02:06 2003

Discussion: The Initiator transmits the SessionType key in the leading login of a session. 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.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #22.1 AuthMethod

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

Reference: iSCSI Standard 11.1, 11.1.4

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

Last Modification: February 23, 2005

Discussion: The authentication methods that can be offered are either those listed here or are vendor-unique methods: KRB5, SPKM1, SPKM2, SRP, CHAP, None. The initiator and target **MUST** implement CHAP. All other authentication methods are optional.

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 value from the list offered by the Testing Station.
- If the DUT originates AuthMethod, verify that CHAP is included in the list.

Possible Problems: None.

The University of New Hampshire
InterOperability Laboratory

Test #23.1 TargetPortalGroupTag

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

Reference: iSCSI Standard 5.3.1, 12.9, need reference about C bit

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

Last Modification: February 23, 2005

Discussion: 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). 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.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #24.1 C bit

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

Reference: iSCSI Standard 5, 5.1, 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: Mon May 19 16:02:40 2003

Discussion: Since some key=value pairs may not fit entirely in a single PDU, the C (continuation) bit is used (both in Login and Text) to indicate that "more follows". 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). 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). 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: None.

*The University of New Hampshire
InterOperability Laboratory*

Test #25.1 Redirect

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

Reference: iSCSI Standard 10.13.5

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

Last Modification: Mon May 19 16:02:49 2003

Discussion: The Status returned in a Login Response indicates the execution status of the Login Phase. The status includes Status-Class and Status-Detail. 0 Status-Class indicates success. A non-zero Status-Class indicates an exception. A Status-Class of 1 means Redirection. 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.

*The University of New Hampshire
InterOperability Laboratory*

Test #26.1 Errors Invalid Keys

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

Reference: iSCSI Standard 12

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

Last Modification: Mon May 19 16:03:01 2003

Discussion: Login/Text Operational Keys are defined for use either by initiator or target. If an initiator were to transmit a key not allowed for its device type, this would indicate a major implementation problem. The device detecting this should reject the 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 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 either transmit a Login Reject PDU, or value=NotUnderstood.

Possible Problems: None.

*The University of New Hampshire
InterOperability Laboratory*

Test #26.2.1 Errors X Keys

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

Reference: iSCSI Standard 12

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

Last Modification: Mon May 19 16:03:08 2003

Discussion: If an iSCSI device does not recognize a vendor specific X key, it should reply with the value 'Not Understood'.

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.

*The University of New Hampshire
InterOperability Laboratory*

Test #26.2.2 Errors X Keys

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

Reference: iSCSI Standard 5.1

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

Last Modification: Mon May 19 16:03:14 2003

Discussion: A key-name can be a string of one or more characters that consist 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:

- The DUT should reject the received key or transmit "NotUnderstood".

Possible Problems: None.

*The University of New Hampshire
InterOperability Laboratory*

Test #26.3.2 Errors Big Values

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

Reference: iSCSI Standard 5.1, 5.2, 6.10

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

Last Modification: April 11, 2006

Discussion: 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). If an iSCSI device recognizes a value as too long, the value should be rejected. If the value is a declared value (as opposed to a negotiated value) no response to the key is required. In this case the responder may choose to terminate the connection since an invalid value has been used. This is an informative test.

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 = WickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTarget

Observable Results:

- The DUT should reject the received key.

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 clause 5.2 of the iSCSI spec do not apply here.

*The University of New Hampshire
InterOperability Laboratory*

Test #26.4 Errors Inquire Value

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

Reference: iSCSI Standard 5.1, 5.2.2

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

Last Modification: Mon May 19 16:03:48 2003

Discussion: The '?' inquire value is not allowed.

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.

Possible Problems: None.

*The University of New Hampshire
InterOperability Laboratory*

Test #27.1 Irrelevant Keys

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

Reference: iSCSI Standard 12

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

Last Modification: February 23, 2005

Discussion: Some keys are defined as Irrelevant during a Discovery Session. These are MaxConnections, InitialR2T, Immediate Data, MaxBurstLength, FirstBurstLength, MaxOutstandingR2T, DataPDUInOrder, and DataSequenceInOrder.

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.

Observable Results:

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

Possible Problems: None.

APPENDICES

Overview:

Test suite appendices are intended to provide additional low-level technical detail pertinent to specific tests contained in this test suite. These appendices often cover topics that are outside of the scope of the standard, and are specific to the methodologies used for performing the measurements in this test suite. Appendix topics may also include discussion regarding a specific interpretation of the standard (for the purposes of this test suite), for cases where a particular specification may appear unclear or otherwise open to multiple interpretations.

Scope:

Test suite appendices are considered informative supplements, and pertain solely to the test definitions and procedures contained in this test suite. The Appendices for this test suite are not yet complete as none of the equipment necessary to perform the tests described in this test suite is available to the UNH-IOL.