iSCSI Consortium
Login Phase Test Suite
For iSCSI Initiators

Version 0.1

Last Update: July 28, 2003

iSCSI Consortium
InterOperability Laboratory
Research Computing Center
University of New Hampshire
http://www.iol.unh.edu

121 Technology Drive Suite 2
Durham, NH 03824-3525
Phone: (603) 862-1908
Fax: (603) 862-4181

MODIFICATION RECORD

1. Currently on Version 0.1. Version 1.0 is awaiting publication of iSCSI RFC

ACKNOWLEDGMENTS

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

David Woolf University of New Hampshire

INTRODUCTION

Overview

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 suite of tests has been developed to help implementers evaluate the functioning of their iSCSI products. The tests do not determine if a product conforms to the iSCSI draft standard, nor are they purely interoperability tests. Rather, they provide one method to isolate problems within an iSCSI device. Successful completion of all tests contained in this suite does not guarantee that the tested device will operate with other iSCSI devices. However, combined with satisfactory operation in the IOL's semi-production environment, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function well in most multivendor iSCSI environments.

Organization of Tests

The tests contained in this document are organized to simplify the identification of information related to a test and to facilitate in the actual testing process. Each test contains an identification section that describes the test and provides cross reference information. The detailed section discusses the background information and specifies how the test is to be performed. Tests are grouped in order to reduce setup time in the lab environment. Each test contains the following information:

Test Label

The Label associated with each test is a title that is used to refer to the test. The attached number is an internal reference number dealing with an internal reference to the test.

Purpose

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

References

The references section lists cross references to the iSCSI draft standard and other documentation that might be helpful in understanding and evaluating the test and results.

Resource Requirements

The requirements section specifies the software, hardware, and test equipment that will be needed to perform the test. The items contained in this section are special test devices,

software that must reside on the DUT, or other facilities which may not be available on all devices.

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.

Test Setup
The setup section describes in detail the configuration of the test environment and includes a block diagram for clarification as well as information such as the interconnection of devices, what monitoring equipment should capture, what the generation equipment should send, and any other configuration information vital to carrying out the test. Small changes in the configuration should be included in the test procedure.

Procedure
The procedure section of the test description contains the step-by-step instructions for carrying out the test. It provides a cookbook approach to testing, and will often be interspersed with observable results.

Observable Results
The observable results section lists observables that can be examined by the tester to verify that the DUT is operating properly. When multiple values are possible for an observable, this section provides a short discussion on how to interpret them. Note that complete delineation between the observables in the **Procedure** and **Observable Results** is virtually impossible. As such a careful note should be made of the requirements in both sections. In certain cases, it may be necessary to modify certain steps in the **Procedure** section while doing the actual tests so as to be able to perform the tests. In such cases, the modifications will be noted in the summary report.

Possible Problems
This section provides some clues to look for if the test does not yield the expected results.

REFERENCES

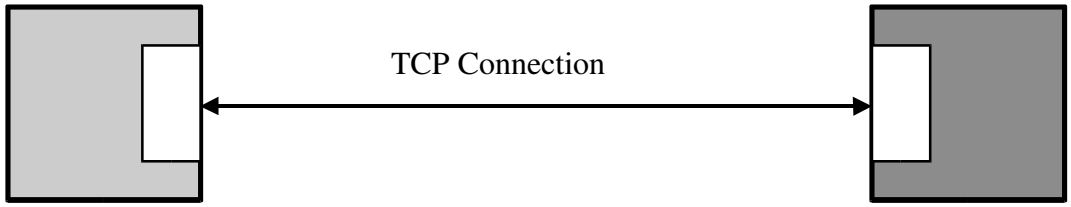
The following documents are referenced in this text:

IETF IPS Working Group iSCSI draft 20

TEST SETUPS

The following test setups are used in this test suite:

Test Setup 1:



Testing Station/ Monitor DUT

TABLE OF CONTENTS

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: 3.5.3.2; 10.12.7; 10.12.4; 10.12.4.1; 10.12.2; 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: Mon May 19 15:45:48 2003

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 within the Login phase MUST carry the same CID. The target MUST use the value presented with the first Login Request. A Login Requests within the Login Phase MUST carry the same Version-max. All Login requests within the Logn Phase MUST carry the same Version-min. The Version number of the current draft is 0x00. As such, all devices MUST carry version 0x00 for both Version-min and Version-max. 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 FullFeaturePhase 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.
·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 the Initiator offers an InitiatorTaskTag in the first Login Request PDU.
·Verify that the Initiator does not change this in the course of a standard Login.
·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 field was constant in all Login Requests.
·Verify that in the login led by the DUT, the Version Min field was constant in all Login Requests.
·Verify that the DUT uses the version number for the current draft 0x00. This may vary depending on what notation is adopted for differentiating between drafts.
·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 the Type field, the Naming Authority field, and Qualifier field are all valid according to the rules discussed above.
·Verify that throughout the login phase the DUT does not use '?' as a value, indicating an inquiry. This is an invalid value.

Possible Problems: None

Test #2.1 TSIH

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

Reference: 3.2.3, 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: Mon May 19 15:46:02 2003

Discussion: For a new session, the TSIH is zero. As part of the response, the target generates a TSIH.TSIH must be set in the first Login Request. The reserved value 0 MUST be used on the first connection for a new session. Otherwise, the TSIH sent by the target at the conclusion of the successful login of the first connection for this session MUST be used. The TSIH identifies to the target the associated existing session for this new connection. 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 that 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.

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 final Login Response with the TSIH field set. Every other Login Response PDU the Testing Station sends will have a TSIH of 0.

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

Possible Problems: None

Test #2.2 TSIH

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

Reference: 10.12.6, 10.13.3, 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 15:46:22 2003

Discussion: For a new session, the TSIH is zero. As part of the response, the target generates a TSIH.TSIH must be set in the first Login Request. The reserved value 0 MUST be used on the first connection for a new session. Otherwise, the TSIH sent by the target at the conclusion of the successful login of the first connection for this session MUST be used. The TSIH identifies to the target the associated existing session for this new connection. 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 that 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.

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

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

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

Possible Problems: None

Test #2.3 TSIH

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

Reference: 10.12.6, 10.13.3, 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 15:46:29 2003

Discussion: For a new session, the TSIH is zero. As part of the response, the target generates a TSIH.TSIH must be set in the first Login Request. The reserved value 0 MUST be used on the first connection for a new session. Otherwise, the TSIH sent by the target at the conclusion of the successful login of the first connection for this session MUST be used. The TSIH identifies to the target the associated existing session for this new connection. 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 that 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.

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 TSIH in the Final Login Respnonse.
·If the device supports multiple connections, allow it to add another 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 Login Request of the new connection.

Possible Problems: None

Test #3.1 T-Bit

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

Reference: 10.12.1

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

Last Modification: Mon May 19 15:46:38 2003

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 FullFeaturePhase, 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
·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.
·Verify that the initiator does not change its value of CSG from the previous 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 the T=1 and NSG=FullFeaturePhase from the DUT, the Testing Station transmits a Login Final Response.Verify that the DUT moves into FullFeaturePhase operation by looking for SCSI Commands.

Possible Problems: None

Test #3.2 T-Bit

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

Reference: 10.12.1

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

Last Modification: Mon May 19 15:46:47 2003

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 FullFeaturePhase, 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
·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 and no parameters included for negotiation in this login response PDU.
·Wait for the initiator to transmit another login request.
·Verify that the initiator does not change its value of CSG from the previous 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. Then 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 T=1 and NSG=FullFeaturePhase, that the next Login Response from the target with T=1 is the final Login Response, and both devices move into the FullFeaturePhase.

Possible Problems: None

Test #4.1 StatSN

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

Reference: 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 15:48:25 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 response with StatSN set to 0.

Observable Results:
·Verify that the DUT initializes ExpStatSN to one greater than the value provided in the StatSN from the Testing Station.

Possible Problems: None

Test #4.2 StatSN

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

Reference: 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 one greater than the value provided in the StatSN from the Testing Station.

Possible Problems: None

Test #5.1 ExpStatSN

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

Reference: 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: Device may not attempt a connection restart, or may only do so if the connection is terminated while in the Full Feature Phase.

Test #6.1 Negotiate Once

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

Reference: 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 performed a Standard Login.

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

Possible Problems: None.

Test #6.2 Negotiate Once

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

Reference: 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 ImmeditateData key twice.

Possible Problems: Some devices may choose to not 'detect' the occurance 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

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: 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 ImmeditateData key twice.

Possible Problems: Some devices may choose to not 'detect' the occurance 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

Test #7.1 Login Request

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

Reference: 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 15:49:43 2003

Discussion: The initial login request of any connection MUST include the InitiatorName key=value pair. The initial login request of the first connection of a session MAY also include the SessionType key=value pair. For any connection within a session whose type is not "Discovery", the first login request MUST also include the TargetName key=value pair. The Login Phase MAY include a SecurityNegotiation stage and a LoginOperationalNegotiation stage and MUST include at least one of them, but the included stage MAY be empty except for the mandatory names. 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:
· Wait for the initiator to connect to the Testing Station and perform a Standard Login.

Observable Results:
·Verify that the initiator performs an Initial Login Request.
·Verify that the device does not move directly into the Full Feature Phase.
·Verify that the device does not attempt to go to Security stage after entering LoginOperational.
·Verify that the Initial Login Request includes 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.
·Verify that the Initial Login Request includes the InitiatorName key and and TargetName key.

Possible Problems: None.

Test #7.2: Login Request

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

Reference: 5.2.1, 5.3.3, 12.1

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

Last Modification: Mon May 19 15:52:20 2003

Discussion: Operational parameter negotiation MAY involve several Login request-response exchanges started and terminated by the initiator. The initiator MUST indicate its intent to terminate the negotiation by setting the T bit to 1; the target sets the T bit to 1 on the last response. 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 OperationalParameter negotiation.
· The Testing Station should transmit a Login Response PDU with any keys offered by the DUT an 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: None.

Test #7.3: Login Request

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

Reference: 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: Mon May 19 15:57:20 2003

Discussion: Operational parameter negotiation MAY involve several Login request-response exchanges started and terminated by the initiator. The initiator MUST indicate its intent to terminate the negotiation by setting the T bit to 1; the target sets the T bit to 1 on the last response. 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 Security Negotiation. The DUT is expected to offer some key=value pair for AuthMethod.
· The Testing Station should transmit a Login Response PDU with the AuthMethod key and a value not offered by the DUT.

Observable Results:
· Verify that the DUT as the requester recognizes this as a failed negotiation and proceeds by dropping the connection.

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

Test #7.4: Login Request

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

Reference: 5.2.2, 5.3.3, 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: Operational parameter negotiation MAY involve several Login request-response exchanges started and terminated by the initiator. The initiator MUST indicate its intent to terminate the negotiation by setting the T bit to 1; the target sets the T bit to 1 on the last response. 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.
· Negotiate ImmediateData 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.

Test #7.5: Login Request

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

Reference: 5.2.2, 5.3.3, 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: Operational parameter negotiation MAY involve several Login request-response exchanges started and terminated by the initiator. The initiator MUST indicate its intent to terminate the negotiation by setting the T bit to 1; the target sets the T bit to 1 on the last response. 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 the following key=value pair: ImmediateData=Ok.

Observable Results:
· Verify that the device responds with the key-value pair ImmediateData=Reject or selects an admissible value.

Possible Problems: None.

Test #7.6: Login Request

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

Reference: 5.2, 5.3.3, 12

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

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

Discussion: Operational parameter negotiation MAY involve several Login request-response exchanges started and terminated by the initiator. The initiator MUST indicate its intent to terminate the negotiation by setting the T bit to 1; the target sets the T bit to 1 on the last response. 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.
· 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 Invalid PDU

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

Reference: 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 the initiator receives any PDU except a Login response, it MUST immediately terminate the connection.

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

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

Observable Results:
· Verify that the device terminates the connection.

Possible Problems: None.

Test #9.1 Header and Data Digests

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

Reference: 12.1

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

Last Modification: Mon May 19 15:58:17 2003

Discussion: CRC32C and None must be offered as options for either Header or Data Digest. The generator polynomial is 0x11edc6f41. 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 device attempts a Header or Data Digest negotiation, it offers CRC32C or none as options.

Possible Problems: None.

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: 12.1

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

Last Modification: Mon May 19 15:58:23 2003

Discussion: CRC32C and none are the only options that a device may offer for either Header or Data Digest. The generator polynomial is 0x11edc6f41. Proprietary algorithms may be listed. Even if the response will be 'None' the DUT must transmit a response.

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.

Test #10.1 MaxConnections

Purpose: To verify that the DUT properly negotiates MaxConnections.

Reference: 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.

Test #11.1 Initiator Name Target Name

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

Reference: 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. TargetName MUST not be redeclared within the login phase. 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 Initiator key enables the initiator to identify itself to the remote endpoint. InitiatorName 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.

Test #12.1 Marker Negotiation

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

Reference: 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: Mon May 19 15:59:02 2003

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 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 OFMarker=Yes key to the Device Under Test.
· Transmit a response with the IFMarker=Yes key to the Device Under Test.
· Transmit a response with the OFMarkInt range of 1~65535.
· Transmit a response with the IFMarkInt range of 1~65535.

Observable Results:
· Verify that the response to both OFMarker and IFMarker is Yes|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'. 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'.

Possible Problems: None.

Test #13.1 Boolean Negotiation

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

Reference: 5.2.2, 12.10, 12.11, 12.18, 12.19, 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 15:59:21 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. ImmediateData 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 InitialR2T=No key.
· Transmit a response with the ImmediateData=Yes key.
· Transmit a response with the DataPDUInOrder=No key.
· Transmit a response with the DataSequenceInOrder=No key.

Observable Results:
· Verify that the DUT responds to and supports the InitialR2T, ImmediateData, DataPDUInOrder, and DataSequenceInOrder keys during the Login phase. Responses are required in all cases.

· Verify that all values offered by the device in response begin with capital letters.

Possible Problems: None.

Test #13.2 Legal Boolean Negotiation

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

Reference: 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 (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. ImmediateData 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 FullFeaturePhase.
· If the InitialR2T key is received with a value of No, return Yes.Proceed through Login to the FullFeaturePhase.
· If the DataPDUInOrder key is received with a value of Yes, do not return the key. Proceed through Login to the FullFeaturePhase.
· If the DataPDUInOrder key is received with a value of No, return Yes.Proceed through Login to the FullFeaturePhase.

· If the DataSequenceInOrder key is received with a value of Yes, do not return the key. Proceed through Login to the FullFeaturePhase.
· If the DataSequenceInOrder key is received with a value of No, return Yes.Proceed through Login to the FullFeaturePhase.
· If the ImmediateData key is received with a value of No, do not return the key. Proceed through login to FullFeaturePhase.
· If the ImmediateData key is received with a value of Yes, respond with No. Proceed through login to FullFeaturePhase.

Observable Results:
· Verify that the DUT does not detect a Negotiation Failure in any of these cases, and does not terminate the connection.
· Verify that the iSCSI intiator initiates SCSI traffic, to show that it has completed Login and entered Full Feature Phase.

Possible Problems: None.

Test #13.3 Illegal Boolean Negotiation

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

Reference: 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. ImmediateData 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 FullFeaturePhase.
· If the InitialR2T key is received with a value of Yes, return No.Proceed through Login to the FullFeaturePhase.
· If the DataPDUInOrder key is received with a value of No, do not return the key. Proceed through Login to the FullFeaturePhase.
· If the DataPDUInOrder key is received with a value of Yes, return No.Proceed through Login to the FullFeaturePhase.

· If the DataSequenceInOrder key is received with a value of No, do not return the key. Proceed through Login to the FullFeaturePhase.
· If the DataSequenceInOrder key is received with a value of Yes, return No.Proceed through Login to the FullFeaturePhase.
· If the ImmediateData key is received with a value of Yes, do not return the key. Proceed through login to FullFeaturePhase.
· If the ImmediateData key is received with a value of No, respond with Yes. Proceed through login to FullFeaturePhase.

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.

Test #14.1 MaxRecvDataSegmentLength

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

Reference: 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.

Test #15.1 MaxBurstLength

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

Reference: 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 MaxBurstLengthe 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.

Test #16.1 FirstBurstLength

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

Reference: 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.

Test #16.2 FirstBurstLength

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

Reference: 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 ImmediateData 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 ImmediateData and InitialR2T that make FirstBurstLength relevant, this item is not testable.

Test #16.3 FirstBurstLength

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

Reference: 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.

Test #17.1 DefaultTime2Retain

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

Reference: 12.16

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

Last Modification: Mon May 19 16:00:29 2003

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.

Observable Results:
· Verify that a device, which offers the DefaultTime2Retain key, only does so in the leading connection, and that the values it presents fall between 0 - 3600.
· Verify that the response value offered by the DUT is less than or equal to the value offered by the testing station.

Possible Problems: None.

Test #18.1 DefaultTime2Wait

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

Reference: 12.15

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

Last Modification: Mon May 19 16:00:37 2003

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.

Observable Results:
· Verify that the device responds to the DefaultTime2Wait key with a value less than or equal to the value offered by the Testing Station.
· If the DUT is the originator of the DefaultTime2Wait key that it presents a range of values with a ~ between 0 and 3600.

Possible Problems: None.

Test #19.1 MaxOutstandingR2T

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

Reference: 12.17

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

Last Modification: Mon May 19 16:00:47 2003

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.

Observable Results:
· Verify that a device, which uses the MaxOutstandingR2T key, only does so in the leading connection of a session, and that the values it presents fall between 1 - 65535.

Possible Problems: None.

Test #20.1 ErrorRecoveryLevel

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

Reference: 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 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 initiator 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.

Test #21.1 SessionType

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

Reference: 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.

Test #22.1 AuthMethod

Purpose: To verify that the DUT supports any AuthMethod, it supports CHAP.

Reference: 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: Mon May 19 16:02:16 2003

Discussion: The authentication methods that can be used (appear in the list-of-values) 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, None.
·

Observable Results:
· Verify that the DUT chooses CHAP or none.
· If the device offers an AuthMethod, verify that CHAP is included in the list.

Possible Problems: None.

Test #23.1 TargetPortalGroupTag

Purpose: To verify that the DUT does not require the presence of the TargetPortalGroupTag in the first Login Response it receives.

Reference: 5.3.1, 12.9

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

Last Modification: Tue May 27 09:54:12 2003

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.
· In the first Login Response the Testing Station should not offer the TargetPortalGroupTag key.
· 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.

Possible Problems: None.

Test #24.1 C bit

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

Reference: 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 FullFeaturePhase.

Possible Problems: None.

Test #25.1 Redirect

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

Reference: 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. In this case, Status-Class is sufficient for a simple initiator to use when handling exceptions, without having to look at the Status-Detail. The Status-Detail allows finer-grained exception handling for more sophisticated initiators and for better information for logging. Redirection indicates that the initiator must take further action to complete the request. This is usually due to the target moving to a different address. All of the redirection status class responses MUST return one or more text key parameters of the type "TargetAddress", which indicates the target's new address.

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

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

Observable Results:
· Verify that the DUT disconnects and then reconnects to the new TargetAddress.

Possible Problems: None.

Test #26.1 Errors Invalid Keys

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

Reference: 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 was 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 PDY, or value=NotUnderstood.

Possible Problems: None.

Test #26.2.1 Errors X Keys

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

Reference: 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.

Test #26.2.2 Errors X Keys

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

Reference: 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

Observable Results:
· The DUT should reject the received key or transmit "NotUnderstood'.

Possible Problems: None.

Test #26.3.1 Errors Big Values

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

Reference: 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 16:03:31 2003

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).

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

Procedure:
· Connect the Testing Station to the DUT and begin a standard login.
· Transmit a Login Response with the following keys=value pair:
ImmediateData=NoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoN
oNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNo
NoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNo
NoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNo
NoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNoNo.

Observable Results:
· The DUT should reject the received key.The DUT may also select an admissible value.

Possible Problems: None.

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: 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 16:03:37 2003

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).

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 = WickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTargetWickedReliableTarget

Observable Results:
· The DUT should reject the received key.

Possible Problems: None.

Test #26.4 Errors Inquire Value

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

Reference: 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 no longer 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 vlaue.

Possible Problems: None.

Test #27.1 Irrelevant Keys

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

Reference: 12

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

Last Modification: Tue May 27 14:13:53 2003

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

Observable Results:
· The DUT should not drop the connection. The DUT is expected to respond to the keys with either a valid response value, or Irrelevant.

Possible Problems: None.