iSCSI Consortium

Login Phase Test Suite
For iSCSI Targets

Version 0.1

Last Update: June 23, 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:

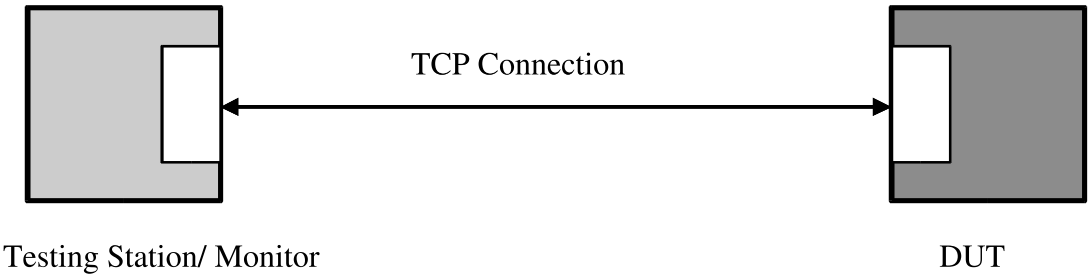

TABLE OF CONTENTS

MODIFICATION RECORD **2**
ACKNOWLEDGMENTS **3**
INTRODUCTION **4**
REFERENCES **6**
TEST SETUPS **7**
TABLE OF CONTENTS **8**
Test #1.1: Standard Login **10**
Test #1.2: Standard Login **13**
Test #2.1: CmdSN **14**
Test #3.1: Version Active **15**
Test #4.1: T Bit **16**
Test #4.2: T Bit **17**
Test #4.3: T Bit **18**
Test #4.4: T Bit **19**
Test #5.1: ExpStatSN **20**
Test #6.1: Negotiate Once **21**
Test #6.2: Negotiate Once **22**
Test #6.3: Negotiate Once **23**
Test #6.4: Negotiate Once **24**
Test #6.5: Negotiate Once **25**
Test #7.1: Login Partial Response **26**
Test #7.2: Login Partial Response **27**
Test #7.3: Login Partial Response **28**
Test #7.4: Login Partial Response **29**
Test #7.5.1: Login Partial Response **30**
Test #7.5.2: Login Partial Response **31**
Test #7.6: Login Partial Response **32**
Test #8.1: Status Detail **33**
Test #9.1: Invalid PDU **34**
Test #9.2: Invalid PDU **35**
Test #10.1: Parameter Names **36**
Test #11.1: AuthMethod **37**
Test #12.1: Header and Data Digest **38**
Test #12.2: Header and Data Digest **39**
Test #13.1: MaxConnections **40**

Test #14.1: TargetAlias **41**
Test #15.1: Marker Negotiation **42**
Test #16.1: FirstBurstLength **43**
Test #16.2: FirstBurstLength **44**
Test #16.3: FirstBurstLength **45**
Test #16.4: FirstBurstLength **46**
Test #17.1: SessionType **47**
Test #18.1: C bit **48**
Test #19.1 Errors Invalid Keys **50**
Test #19.2.1 Errors X Keys **51**
Test #19.2.2 Errors X Keys **52**
Test #19.3.1 Errors Big Values **53**
Test #19.3.2 Errors Big Values **54**
Test #19.4 Errors Inquire Value **55**
Test #20.1: TargetPortalGroupTag Normal **56**
Test #21.1: Irrelevant Keys **57**

Test #1.1: Standard Login

Purpose: To see if the DUT properly uses the following PDU fields and key=value pairs during the Login Phase: Command Code, TSIH, CID, CmdSN, StatSN, TargetPortalGroupTag, Initial R2T, BiDiInitialR2T, Immediate Data, MaxRecvDataSegmentLength, MaxBurstSize, FirstBurstSize, DefaultTime2Wait, DefaultTime2Retain, MaxOutstandingR2T, DataPDUInOrder, DataSequenceInOrder, ErrorRecoveryLevel. To verify that the DUT includes the proper information in its Final Response.

Reference: 10.13; 10.13.3, 3.2.3; 10.12.7; 10.12.8; 10.13.4; 12.9; 12.10; 12.11; 12.12; 12.13; 12.14; 12.15; 12.16; 12.17; 12.18; 12.19; 11.20; 4.3.1, 4.3.2.

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

Last Modification: Mon May 19 16:18:43 2003

Discussion: The command code for an iSCSI login response is 0x23. 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. For a new session the TSIH is zero. As part of the response, the target generates a TSIH. 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. CmdSN is either the initial command sequence number of a session (for the first Login request of a session - the "leading" login), or the command sequence number in the command stream if the login is for a new connection in an existing session. As an example if the Login ia on a leading connection, and if the leading login carries the CmdSN 123, all other login requests in the same login phase carry the CmdSN 123 and the first non-immediate command in FullFeaturePhase also carries the CmdSN 123.If the login request is a leading login request, the target MUST use the value presented in CmdSN as the target value for ExpCmdSN. 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.

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. The Initial R2T key is used to turn off the default use of R2T. This parameter can only be used in the leading connection. The ImmediateData key is used to negotiate support for immediate and unsolicited data. This parameter can only be used in the leading connection. The MaxRecvDataSegmentLength key is used to declare the maximum data segment length in bytes and must be between 512 and 2^24-1. MaxBurstLength key can only be used in the Leading Login of a connection. 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. 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. FirstBurstLength must fall in a range between 512 and 2^24-1. The DefaultTime2Wait can only be used in the leading connection and must be a number from 0 - 3600. The DefaultTime2Retain can only be used in the leading connection and must be a number from 0 - 3600. The MaxOutstandingR2T key can only be used in the leading login of a connection and must be a number from 1 - 65535. The DataPDUInOrder key has a yes|no value and can only be used in the Leading Login of a connection. The DataSequenceInOrder key has a yes|no value and can only be used in the Leading Login of a connection. The ErrorRecoveryLevel key can only be used in in the LeadingLogin of a connection and must have a value between 0 and 2. Both initiator and target send this key. The minimum of the two values is selected. A Login Initial Request and a Login Final response is mandatory before entering the Full Feature Phase. The Login phase MUST include at least 1 of either the SecurityNegotiation Stage or the LoginOperationalNegotiation stage, and MAY include either. If both Security and LoginOperational stages are used, then Security must precede LoginOperational. The initial login request includes: protocol version supported, session and connection ID, the negotiation stage that the initiator is ready to enter. If a target chooses to respond to a login request received with the T bit set and NSG is set to FullFeaturePhase, with a Login Final Response, that login final response includes a protocol version and a session ID.

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

Procedure:

· Connect the Testing Station to the Target.
· The Testing Station should attempt to perform a standard login with the target. Use a value of 123 for CmdSN. Attempt to negotiate the following keys: InitialR2T,

ImmediateData, MaxRecvDataSegmentLength, MaxBurstLength, FirstBurstLength,DefaultTime2Retain, DefaultTime2Wait, MaxOutstandingR2T, DataPDUInOrder,DataSequenceInOrder, ErrorRecoveryLevel.

Observable Results:

· Verify that in each login response that the DUT uses the 0x23 command code.
· Verify that the target sets the TSIH field only in the final login response, (i.e. the TSIH equals 0 in every response but the final login response). Verify that it is formatted properly.
· Verify that the target uses the CID provided in the first Login Request
· Verify that a target uses the provided value for CmdSN by checking ExpCmdSN.
· Verify that a target sets StatSN and increments it with each Login response.
· Verify that the DUT includes the TargetPortalGroupTag key=value pair in its first Login Response PDU.
· Verify that the DUT responds to and supports the InitialR2T key during the Login phase. The DUTs response should begin with a capital letter.
· Verify that the DUT responds to and supports the ImmediateData key during the Login phase. The DUTs response should begin with a capital letter.
· Verify that the DUT supports the MaxRecvDataSegmentLength key during the Login phase, no response is expected since this is a declarative key, but the DUT is expected to accept the key.
· Verify that the MaxBurstLength key is responded to properly by the device under test. Verify the value requested falls within the range specified
· Verify that the FirstBurstLength key is transmitted and responded to properly by the device under test. Verify the value requested falls within the value negotiated for MaxBurstLength.
· Verify that a device, which uses the DefaultTime2Retain key, only presents values between 0 - 3600. Verify that a device only uses this key in the leading connection.
· Verify that a device, which uses the DefaultTime2Wait key, only presents values between 0 - 3600. Verify that a device only uses this key in the leading connection.
· Verify that a device, which uses the MaxOutstandingR2T key, only does so in the leading login of a connection, and that the values it presents fall between 1 - 65535.
· Verify that a device, which uses the DataPDUInOrder key, only does so in the leading login of a connection, and that formats the key=value pair properly.
· Verify that a device, which uses the DataSequenceInOrder key, only does so in the leading login of a connection, and that formats the key=value pair properly.
· Verify that a device, which uses the ErrorRecoveryLevel key, only does so in the leading login of a connection, and that formats the key=value pair properly.
· Verify that the Target transmits a Final Login Response (T=1 , NSG=FullFeaturePhase). Verify that this response includes a protocol version and a session ID.

· Verify that a value of '?' is not used during negotiation to indicate 'inquiry'.

Possible Problems: None.

Test #1.2: Standard Login

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

Reference: 10.12; 10.12.4, 10.13.1; 10.13.2.

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

Last Modification: Mon May 19 16:18:57 2003

Discussion: The Login Phase consists of a sequence of Login requests and responses that carry the same Initiator Task Tag. The version number of the current draft is 0x00. As such, all devices MUST carry version 0x00 for both Version-min and Version-max. The target MUST use the value presented with the first login request. The Version-active field indicates the highest version supported by the target and initiator. If the target does not support a version within the range specified by the initiator, the target rejects the login and this field indicates the lowest version supported by the target.All Login responses within the Login Phase MUST carry the same Version-active.

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

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

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

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

Possible Problems: None.

Test #2.1: CmdSN

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

Reference: 10.12.8

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

Last Modification: Mon May 19 16:19:07 2003

Discussion: CmdSN is either the initial command sequence number of a session (for the first Login request of a session - the "leading" login), or the command sequence number in the command stream if the login is for a new connection in an existing session.If the login request is a leading login request, the target MUST use the value presented in CmdSN as the target value for ExpCmdSN.

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

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

Observable Results:
· Verify that a target uses the provided value for CmdSN as the target value for ExpCmdSN.

Possible Problems: None.

Test #3.1: Version Active

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

Reference: 10.12.4, 10.13.2

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

Last Modification: Mon May 19 16:19:18 2003

Discussion: The version number of the current draft is 0x00. As such, all devices MUST carry version 0x00 for both Version-min and Version-max. Version Active indicates the highest version supported by the target and initiator. If the target does not support a version within the range specified by the initiator, the target rejects the login and this field indicates the lowest version supported by the target. All Login responses within the Login Phase MUST carry the same Version-active.

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

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

Observable Results:
· Verify that the Login Response is a reject and verify that the Version Active field contains the targets lowest supported version, 0x00.

Possible Problems: None.

Test #4.1: T Bit

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

Reference: 5.3, 10.13.6

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

Last Modification: Mon May 19 16:19:28 2003

Discussion: Targets MUST NOT submit parameters that require an additional initiator login request in a login response with the T bit set to 1. A login response with a T bit set to 1 MUST NOT contain key=value pairs that may require additional answers from the initiator within the same stage. If the status class is 0, the T bit MUST NOT be set to 1 if the T bit in the request was set to 0.

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

Procedure:
· Connect the Testing Station to the DUT.
· The Testing Station should transmit a second Login Request with T=1, CSG=Security Negotiation, NSG=Operational Parameter Negotiation. Repeat until the target responds with T=1.
· The Testing Station should now send a Login Request indicating CSG=Operational Parameter Negotiation, T=0. The Testing Station should transmit as many Login PDUs as possible, each offering operational parameters, to expand the time that the devices spend in negotiation.

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

Possible Problems: None.

Test #4.2: T Bit

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

Reference: 10.12.1, 10.12.3

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

Last Modification: Mon May 19 16:19:40 2003

Discussion: If set to 1 the T bit indicates that the initiator is ready to transit to the next stage. If the T bit is set to 1 and NSG is FullFeaturePhase, then this also indicates that the initiator is ready for the Final Login Response. The next stage value (NSG) is only valid when the T bit is 1; otherwise, it is reserved.

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

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

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

Possible Problems: None.

Test #4.3: T Bit

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

Reference: 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 16:19:57 2003

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

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

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

Observable Results:
· Verify that the DUT can follow the specified paths.

Possible Problems: None.

Test #4.4: T Bit

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

Reference: 5.3.3

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

Last Modification: Tue May 20 09:25:07 2003

Discussion: If the target responds to a Login request that has the T bit set to 1 with a Login response that has the T bit set to 0, the initiator should keep sending the Login request (even empty) with the T bit set to 1, while it still wants to switch stage, until it receives the Login Response that has the T bit set to 1 or it receives a key that requires it to set the T bit to 0.

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

Procedure:
· Connect the Testing Station to the DUT.
· Once on the OperationalNegotiation stage the Testing Station should send a Login Request PDU with T=0 and no parameters offered for negotiation, wait for a Login Response to be received. The DUT should not treat this an error. Repeat 5 times.
· Transmit a Login Response with T=1, proceed to FullFeaturePhase.

Observable Results:
· Verify that the DUT does not treat the received Request PDUs as errors.

Possible Problems: None.

Test #5.1: ExpStatSN

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

Reference: 10.12.9

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

Last Modification: Tue May 20 09:25:16 2003

Discussion: The ExpStatSN field is valid only of the Login request restarts a connection, in which case the TSIH is not zero. Otherwise this field is reserved.

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

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

Observable Results:
· Verify that the DUT ignores the ExpStatSN field since connection reinstatement is not occuring.

Possible Problems: None.

Test #6.1: Negotiate Once

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

Reference: 5.1, 5.3

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

Last Modification: Tue May 20 09:25:26 2003

Discussion: Every key=value pair, including the last or only pair in a LTDS, MUST be followed by one null (0x00) delimiter. Neither the initiator nor the target should attempt to declare or negotiate a parameter more than once during login except for responses to specific keys that explicitly allow repeated key declarations (e.g., TargetAddress). If an attempt to re-negotiate/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error). If detected by the initiator, the initiator MUST drop the connection.

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

Procedure:
· Connect the Testing Station to the DUT.
· Perform a standard login.

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

Possible Problems: None.

Test #6.2: Negotiate Once

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

Reference: 5.3

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

Last Modification: Tue May 20 09:25:32 2003

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

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

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

Observable Results:
· Verify that the device transmits a Login Reject of reason code initiator error.

Possible Problems: None.

Test #6.3: Negotiate Once

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

Reference: 5.3

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

Last Modification: Tue May 20 09:25:37 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/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error). If detected by the initiator, the initiator MUST drop the connection.

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

Procedure:

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

Observable Results:

- Verify that the device transmits a Login Reject.

Possible Problems: None.

Test #6.4: Negotiate Once

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

Reference: 5.3

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

Last Modification: Tue May 20 09:25:44 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/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error). If detected by the initiator, the initiator MUST drop the connection.

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

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

Observable Results:
· Verify that the device transmits a Login Reject.

Possible Problems: None.

Test #6.5: Negotiate Once

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

Reference: 5.3

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

Last Modification: Tue May 20 09:28:51 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/redeclare parameters not specifically allowed is detected by the target, the target MUST respond with Login reject (initiator error). If detected by the initiator, the initiator MUST drop the connection.

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

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

Observable Results:
· Verify that the device transmits a Login Reject.

Possible Problems: None.

Test #7.1: Login Partial Response

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

Reference: 5.3.1

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

Last Modification: Tue May 20 09:26:04 2003

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

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

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

Observable Results:
· Verify that the login partial response contains the protocol version supported and either security or operational parameters.

Possible Problems: None.

Test #7.2: Login Partial Response

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

Reference: 5.3.2

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

Last Modification: Tue May 20 09:29:15 2003

Discussion: During Login initiator may send a login request with an ordered list of the options it supports (authentication algorithm). The options are listed in the initiator's order of preference. The target MUST reply with the first option in the list it supports and is allowed to use for the specific initiator unless it does not support any in which case it MUST answer with "Reject"

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

Procedure:
· Connect the Testing Station to the DUT.
· The Testing Station should transmit a Login Request PDU with the following key=value pair: DataDigest=CRC32C,peanutbutter,jelly,sandwich,none.

Observable Results:
· Verify that the device responds with the first value it supports and ignores all other values.

Possible Problems: None.

Test #7.3: Login Partial Response

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

Reference: 5.3.2

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

Last Modification: Tue May 20 09:29:22 2003

Discussion: During Login initiator may send a login request with an ordered list of the options it supports (authentication algorithm). The options are listed in the initiator's order of preference. The target MUST reply with the first option in the list it supports and is allowed to use for the specific initiator unless it does not support any in which case it MUST answer with "Reject"

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

Procedure:

· Connect the Testing Station to the DUT.
· The Testing Station should transmit a Login Request PDU with the following key=value pair: AuthMethod = SRP.

Observable Results:

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

Possible Problems: None.

Test #7.4: Login Partial Response

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

Reference: 5.2.2

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

Last Modification: Tue May 20 09:29:28 2003

Discussion: During a simple value negotiation, proposing a value not admissible (e.g., not within the specified bounds) MAY be answered with the constant "Reject" or the acceptor MAY select an admissible value.

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

Procedure:

- Connect the Testing Station to the DUT.
- Negotiate a combination of InitialR2T and ImmediateData such that FirstBurstLength is releveant. The Testing Station should transmit a Login Request PDU with the following key=value pair: FirstBurstLength = 16777216. This value is higher than the maximum value specified for FirstBurstLength.

Observable Results:

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

Possible Problems: If the DUT only supprts InitialR2T=Yes ImmediateData=No, this item is Not Testable.

Test #7.5.1: Login Partial Response

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

Reference: 5.2.2

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

Last Modification: Tue May 20 09:29:36 2003

Discussion: During a simple-value negotiation negotiation, proposing a value not admissible (e.g., not within the specified bounds) MAY be answered with the constant "Reject" or the acceptor MAY select an admissible value.

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

Procedure:
· Connect the Testing Station to the DUT.
· The Testing Station should transmit a Login Request PDU with the following key=value pair: ImmediateData = Ok.

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

Possible Problems: None.

Test #7.5.2: Login Partial Response

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

Reference: 5.2.2

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

Last Modification: Tue May 20 09:29:44 2003

Discussion: During a simple-value negotiation negotiation, proposing a value not admissible (e.g., not within the specified bounds) MAY be answered with the constant "Reject" or the acceptor MAY select an admissible value.

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

Procedure:
· Connect the Testing Station to the DUT.
· The Testing Station should transmit a Login Request PDU with the following key=value pair: DataPDUInOrder = Ok.

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

Possible Problems: None.

Test #7.6: Login Partial Response

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

Reference: 5.2

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

Last Modification: Tue May 20 09:29:52 2003

Discussion: Any key not understood by the acceptor may be ignored by the acceptor without affecting the basic function. However, the answer for a key not understood MUST be key=NotUnderstood. The constants "None", "Reject", "Irrelevant", and "NotUnderstood" are reserved and MUST ONLY be used as described here. Violation of this rule is a protocol error.

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

Procedure:

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

Observable Results:

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

Possible Problems: None.

Test #8.1: Status Detail

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

Reference: 5.3.1, 10.13.5

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

Last Modification: Tue May 20 09:30:03 2003

Discussion: The target can answer a Login Request with a Login Response with Login reject. This is an immeditae rejection from the target that causes the connection to terminate and the session to terminate if this is the first connection of a new session. If the Status Class is not 0, the initiator and target MUST close the TCP connection.

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

Procedure:

- Connect the Testing Station to the DUT.
- The Testing Station transmits a Login Request with the Version Max field set to 1 and the Version Min field set to 4.

Observable Results:

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

Possible Problems: Some targets may not be checking these fields since the draft only specifies a value of 0 for these fields.

Test #9.1: Invalid PDU

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

Reference: 3.2.3, 5.3.1, 10.13.5

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

Last Modification: Tue May 20 09:30:16 2003

Discussion: Once the Login phase has started, if the target receives any PDU except a Login request, it MUST send a Login reject (with Status "invalid during login") and then disconnect. The T bit and the CSG and NSG fields are reserved in a Login Response with Login reject.

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

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

Observable Results:
· Verify that the DUT does not transmit a Login response with a Status Class/ Status Detail of 0x0000, but instead generates a Login Response with an appropriate Status Detail field (0x020B).
· Verify that the T bit, NSG, and CSG all are set to 0.

Possible Problems: None.

Test #9.2: Invalid PDU

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

Reference: 3.2.3

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

Last Modification: Tue May 20 09:30:23 2003

Discussion: A target receiving any PDU except a Login request before the Login phase is started MUST immediately terminate the connection on which the PDU was received.

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

Procedure:
· Connect the Testing Station to the DUT.
· Do not transmit an initial valid Login Request, instead transmit a SCSI Command PDU to the device.

Observable Results:
· Verify that the DUT does not transmit any Login response but instead terminates the connection immediately.

Possible Problems: None.

Test #10.1: Parameter Names

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

Reference: 5.1

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

Last Modification: Tue May 20 09:30:31 2003

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

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

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

Observable Results:
· Verify that all parameter names appear in the format key=value format described above.

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

Possible Problems: None.

Test #11.1: AuthMethod

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

Reference: 11.1

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

Last Modification: Tue May 20 09:30:38 2003

Discussion: The initiator and target MUST implement CHAP. All other authentication methods are OPTIONAL. Private or public extension algorithms MAY also be negotiated for authentication methods. Whenever a private or public extension algorithm is offered, "None" or "CHAP" MUST be listed as an option in order to guarantee interoperability.

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

Procedure:

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

Observable Results:

- If the device offers an AuthMethod, that CHAP is included in the list.
- Verify that the device chooses either CHAP or none.

Possible Problems: None.

Test #12.1: Header and Data Digest

Purpose: To verify that the DUT properly negotiates values for 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: Tue May 20 09:30:48 2003

Discussion: CRC32C and none are the only options that a device must offer for either Header or Data Digest. The generator polynomial is 0x11edc6f41. Proprietary algorithms may be listed with a Yor Y# extension. Support for public or private extension digests is OPTIONAL.

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

Procedure:
· Connect the Testing Station to the DUT.
· Perform a Standard Login

Observable Results:
· Verify if the device attempts a Header or Data Digest negotiation, it offers CRC32C or none as options.

Possible Problems: None.

Test #12.2: Header and Data Digest

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: Tue May 20 09:31:01 2003

Discussion: CRC32C and none are the only options that a device must offer for either Header or Data Digest. The generator polynomial is 0x11edc6f41. Proprietary algorithms may be listed with a Yor Y# extension. Support for public or private extension digests is OPTIONAL.

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

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

Observable Results:
· Verify that the device responds with the value 'None'.

Possible Problems: None.

Test #13.1: MaxConnections

Purpose: To verify that the DUT properly negotiates a value for MaxConnections.

Reference: 12.2

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

Last Modification: Tue May 20 09:31:16 2003

Discussion: MaxConnections can only be negotiated in the leading connection of a session. MaxConnections can range from 1 - 65535.

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

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

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

Possible Problems: None.

Test #14.1: TargetAlias

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

Reference: 12.6

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

Last Modification: Tue May 20 09:31:25 2003

Discussion: If a target has been configured with a human-readable name or description, this name SHOULD be communicated to the initiator during a Login Response PDU if SessionType=Normal (see Section 11.21 SessionType). This string is not used as an identifier, nor is it meant to be used for authentication or authorization decisions. It can be displayed by the initiator's user interface in a list of targets to which it is connected.

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

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

Observable Results:
· Verify that the target communicates the TargetAlias.

Possible Problems: The TargetAlias key is optional for the target to support.

Test #15.1: Marker Negotiation

Purpose: To verify that the DUT properly offers values for 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: Tue May 20 09:31:34 2003

Discussion: The use of markers is negotiable. The initiator and target MAY indicate their readiness to receive and/or send markers during login separately for each connection. The default is No. The OFMarker and IFMarker keys can have values of Yes|No. These keys can only be negotiated during the Login phase. The default value is No, the result function is AND. The OFMarkInt and IFMarkInt keys specify a numerical range of integer values between 1 and 65535. Values specifying a numerical range must be separated by a tidle ~ (0x7e). A device responding to this key must offer an integer value between 1 and 65535 or Reject. The default is 2048. 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. The following example illustrates the use of "Irrelevant": I->T OFMarker=Yes,OFMarkInt=2048~8192 T->I OFMarker=No,OFMarkInt=Irrelevant

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

Procedure:
· Connect the Testing Station to the DUT and perform a standard login.
· Transmit a request with the OFMarker=Yes key to the Device Under Test.
· Transmit a request with the IFMarker=Yes key to the Device Under Test.
· Transmit a request with the OFMarkInt range of 1 to 65535.
· Transmit a request with the IFMarkInt range of 1 to 65535.

Observable Results:
· Verify that the 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 if the response to OFMarker and IFMarker was No.

Possible Problems: None.

Test #16.1: FirstBurstLength

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

Reference: 12.14

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

Last Modification: Tue May 20 09:34:01 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 SCSI command. This covers the immediate data (if any) and the sequence of unsolicited Data-Out PDUs (if any) that follow the command. FirstBurstLength must not exceed MaxBurstLength. A value of zero is not allowed. FirstBurstLength may range from 512 to 2^24-1.

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

Procedure:
· Connect the Testing Station to the DUT and perform a standard login.
· Negotiate a combination of InitialR2T and ImmediateData so that FirstBurstLength is relevant.
· Offer the MaxBurstLength key.
· In the second Login Request transmit a request with the FirstBurstLength key, greater than the MaxBurstLength key.

Observable Results:
· Verify that the FirstBurstLength key is either rejected by the DUT, or the DUT offers avalue for FirstBurstLength that falls within the legal range.

Possible Problems: None.

Test #16.2: FirstBurstLength

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

Reference: 12.14

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

Last Modification: Tue May 20 09:34:12 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 SCSI command. This covers the immediate data (if any) and the sequence of unsolicited Data-Out PDUs (if any) that follow the command. FirstBurstLength must not exceed MaxBurstLength. A value of zero is not allowed. FirstBurstLength may range from 512 to 2^24-1.

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

Procedure:

· Connect the Testing Station to the DUT and perform a standard login.

Observable Results:

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

Possible Problems: If FirstBurstLength is irrelevant due to values negotiated for ImmediateData and InitialR2T this item is Not Testable.

Test #16.3: FirstBurstLength

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

Reference: 12.14

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

Last Modification: Tue May 20 09:34:18 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 SCSI command. This covers the immediate data (if any) and the sequence of unsolicited Data-Out PDUs (if any) that follow the command. FirstBurstLength must not exceed MaxBurstLength. A value of zero is not allowed. FirstBurstLength may range from 512 to 2^24-1.

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

Procedure:
· Connect the Testing Station to the DUT and perform a standard login.
· Negotiate InitialR2T and ImmediateData so that FirstBurstLength is relevant.
· The Testing Station offers a MaxBurstLength key which is less than the default value of FirstBurstLength.

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

Possible Problems: If FirstBurstLength is irrelevant due to values negotiated for ImmediateData and InitialR2T this item is Not Testable.

Test #16.4: FirstBurstLength

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

Reference: 12.14

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

Last Modification: Tue May 20 09:34:24 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 SCSI command. This covers the immediate data (if any) and the sequence of unsolicited Data-Out PDUs (if any) that follow the command. FirstBurstLength must not exceed MaxBurstLength. A value of zero is not allowed. FirstBurstLength may range from 512 to 2^24-1.

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

Procedure:

· Connect the Testing Station to the DUT and perform a standard login.
· Negotiate InitiatlR2T and ImmediateData such that FirstBurstLength is relevant.
· The Testing Station offers a FirstBurstLength key of size greater than the default value for MaxBurstLength, but still a legal value.

Observable Results:

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

Possible Problems: If FirstBurstLength is irrelevant due to values negotiated for ImmediateData and InitialR2T this item is Not Testable.

Test #17.1: SessionType

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

Reference: 12.21

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

Last Modification: Tue May 20 09:34:53 2003

Discussion: The SessionType key can only be used in the Leading Login of a connection, and may have the value of Discovery|Normal. The SessionType key can only be sent be the initiator. The initiator indicates the type of session it wants to create, the target can accept or reject it. Discovery session implies MaxConnections=1 and overrides both the default and an explicit setting.

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

Procedure:
· Connect the Testing Station to the DUT and perform a standard login.
· Transmit a request with the key=value pair SessionType=Discovery.

Observable Results:
· Verify that the DUT accepts or rejects the key=value pair SessionType=Discovery.

Possible Problems: None.

Test #18.1: C bit

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

Reference: 5.1, 5.2, 10.12.2

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

Last Modification: Tue May 20 09:35:06 2003

Discussion: 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. The C bit, when set to 1, indicates that the text (set of key=value pairs) in this Login Request is not complete (it will be continued on subsequent Login Requests); otherwise, it indicates that this Login Request ends a set of key=value pairs. A Login Request with the C bit set to 1 MUST have the T bit set to 0. A target receiving a Text or Login Request with the C bit set to 1 MUST answer with a Text or Login Response with no data segment (DataSegmentLength 0).

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

Procedure:
· Connect the Testing Station to the DUT and begin a standard login.
· Transmit a Login Response to the DUT, with the C bit =1. T bit = 0, and the following keys: X-cbit.ioliscsilab.test-n = 255 bytes of random data. Keys with values for 'n' of 1 - 32 should be included in this request, up to 8192 bytes. The final key = value pair in this request should be 'MaxRecvDataSegment' then the end of the data segment.
· Transmit a second Text Request to the DUT with the C bit = 0, T bit = 1, and the final portion of the request: 'Length=512'.
· Proceed to the Full Feature Phase.
· Transmit a READ command to the DUT. Wait for Data-in PDUs.

Observable Results:
· The DUT should transmit 'NotUnderstood' to the vendor specific keys. The DUT should not disconnect.

· Veridy that the DUT responds the recieved Login Request which had the C bit set to 1, with a Login Response with no data segment.
· Verify that the DUT adheres to the MaxRecvDataSegmentLength declared by the Testing Station when sending Data-Out PDUs.

Possible Problems: None.

Test #19.1 Errors Invalid Keys

Purpose: To verify that the DUT recognizes keys that are invalid for an initiator to transmit.

Reference: 5.3, 12.6, 12.8, 12.9

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

Last Modification: Tue May 20 09:35:22 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.

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

Procedure:
· Connect the Testing Station to the DUT and begin a standard login.
· Transmit a Login Request with the following key=value pairs: TargetAlias=UNHIOL, TargetPortalGroupTag=1, TargetAddress=10.0.0.1:3260,1.

Observable Results:
· The DUT may do any of the following: Transmit key=Reject to each of the keys. Alternately the DUT may transmit key=Reject to each of the keys and terminate the connection. Or the DUT may transmit key=NotUnderstood to each of the keys since they are not being used as specified. The DUT also has the option to ignore this error. This test is included in the test suite for information only.

Possible Problems: This test is included in the test suite for information only.

Test #19.2.1 Errors X Keys

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

Reference: 5.2

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

Last Modification: Tue May 20 09:35:30 2003

Discussion: All keys in this document, except for the X extension formats, MUST be supported by iSCSI initiators and targets when used as specified here. If used as specified, these keys MUST NOT be answered with NotUnderstood. Implementers may introduce new keys by prefixing them with X-followed by their (reversed) domain name, or with new keys registered with IANA prefixing them with X#. If an iSCSI device does not recognize a vendor specific X key, it should reply with the value 'NotUnderstood'.

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

Procedure:

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

Observable Results:

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

Possible Problems: None.

Test #19.2.2 Errors X Keys

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

Reference: 5.2

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

Last Modification: Tue May 20 09:35:36 2003

Discussion: There is a limit of 63 characters on key names. If an iSCSI device recognizes a vendor specific X key as too long, the key should be rejected. Many devices will check to see if the key is understood first, and will respond with 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 and begin a standard login.
· Transmit a Login Request with the following keys=value pair: X-edu.unh.iol-extension-key-which-is-clearly-longer-than-it-ought-to-be-1=test

Observable Results:
· The DUT should either reject the received key since it is too long, and/or terminate the connection. Alternately the DUT could transmit 'NotUnderstood' to the received keys.

Possible Problems: None.

Test #19.3.1 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: Tue May 20 09:35:48 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). If an iSCSI device recognizes a value as too long, the value should be rejected.

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

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

Observable Results:
· The DUT should reject the received value. The DUT may also choose to terminate the connection. The DUT may also choose to reply with an admissible value for the given key.

Possible Problems: None.

Test #19.3.2 Errors Big Values

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: Tue May 20 09:35:54 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). If an iSCSI device recognizes a value as too long, the value should be rejected. If an iSCSI device recognizes a value as too long, the value should be rejected.

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

Procedure:
· Connect the Testing Station to the DUT and begin a standard login.
· Transmit a Login Request with the following keys=value pair: InitiatorAlias = SuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiatorSuperFastInitiator

Observable Results:
· The DUT should terminate the connection.

Possible Problems: None.

Test #19.4 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: Tue May 20 09:36:01 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 Response with the following key=value pair: MaxConnections=?

Observable Results:
· The DUT should reject the received key. The DUT may also choose to terminate the connection. The DUT may also choose to reply with an admissible value for the given key.

Possible Problems: None.

Test #20.1: TargetPortalGroupTag Normal

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

Reference: 5.3, 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 09:51:40 2003

Discussion: 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. 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).

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

Procedure:
· Transmit a Login Request with key=value pair SessionType=Normal to the DUT.

Observable Results:
· Verify that the DUT transmits a Login Response with the TargetPortalGroupTag key included, with a valid value.

Possible Problems: None.

Test #21.1: Irrelevant Keys

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

Reference: 12

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

Last Modification: Mon Jun 23 09:34:39 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:
· Transmit a Login Request with key=value pair SessionType=Discovery to the DUT.
· After receiving a Login Response, transmit a Login Request with the following keys, each with a valid value: MaxConnections=10, InitialR2T=No, ImmediateData=Yes, MaxBurstLength=2**24-1, FirstBurstLength=2**24-1, MaxOutstandingR2T=10, DataPDUInOrder=No, DatasequenceInOrder=No.

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

Possible Problems: None.