Moonv6 Test Suite

IPv6 Firewall Base
Functionality Test Suite

Technical Document

Revision 0.11

IPv6 Consortium
InterOperability Laboratory
Research Computing Center
University of New Hampshire

121 Technology Drive, Suite 2
Durham, NH 03824-3525
Phone: (603) 862-2804
Fax: (603) 862-4181
http://www.iol.unh.edu

TABLE OF CONTENTS

MODIFICATION RECORD

Draft Version Complete	*February 22, 2004*
Version 0.3	*February 24, 2004: Fixed inconsistencies and typos.*
Version 0.5	*Removed Part E from FIR.1.10. Added Group 5, added Part C to FIR 2.3, Added FIR.4.3, FIR.4.4.and FIR.4.5. Added part D to FIR.4.1. Added test FIR.1.12.*
Version 0.6	*Added discussion on interface definition. Added Part E to FIR.1.7 and an extra set of parts to FIR.4.5 (UDP State Inspection).*
Version 0.7	*August 11, 2004: Fixed errors found in Moonv6 Phase II*
Version 0.8	*September 9, 2004: Separated the Base Test plan items here and created a separate Policy test plan.*
Version 0.9	*September 30, 2004: Updated based on comments received.*
Version 0.10	*October 28, 2004: added sections5 and 6…. Based on comments and suggestions received.*
Version 0.11	*Octorber 30, 2004: added comments, changed test order, added test 6.1*

ACKNOWLEDGEMENTS

The University of New Hampshire would like to acknowledge the efforts of the following individuals in the development of this test suite. This test suite belongs to the University of New Hampshire InterOperability Lab and is a collaborative effort of those listed below and the participants of Moonv6. Special thanks to Check Point for the base test items. Special thanks to Cisco and NetScreen for contributing additional test ideas for the base document for the first revision.

Yoni Appel	Check Point Software Technologies
Peter Atanasovski	Agilent Technologies
Alan Bavosa	NetScreen Technologies
Ankur Chadda	University of New Hampshire
Paul Del Fante	Cisco Systems
Eli Ginot	Check Point Software Technologies
Vincent Le May	6Wind
Changming Liu	NetScreen Technologies
Shiva Mittal	Cisco Systems
Jeff Parker	Cisco Systems
Kari Revier	University of New Hampshire
Cathy Rhoades	University of New Hampshire
Benjamin Schultz	University of New Hampshire
Zlata Trhulj	Agilent Technologies
L. Brad Upson	University of New Hampshire
Dennis Vogel	Cisco Systems
Erica Williamsen	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 Internet Protocol, version 6 firewall products. The tests do not determine if a product conforms to the IPv6 specifications, nor are they purely interoperability tests. Rather, they provide one method to isolate problems within a device. Successful completion of all tests contained in this suite does not guarantee that the tested device will interoperate with other IPv6 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 will function well in most multi-vendor IPv6 environments.

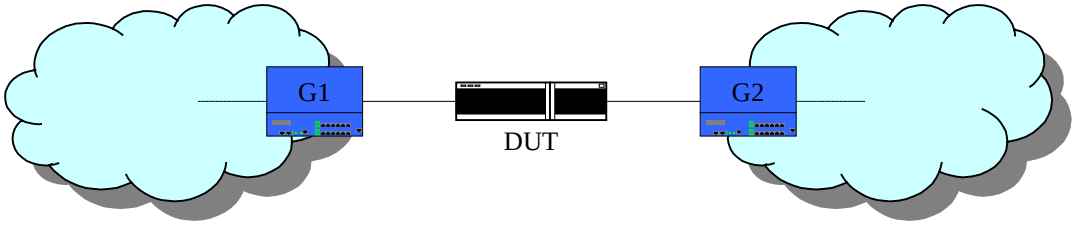
In this test suite, when using interface oriented terms such as "accept...on the interface" or "configure ... on its interface", it is up to the firewall vendor to supply the desired functionality according to the implementation of the DUT. The term "interface" only describes the externally observable behavior, not the specifics of an internal configuration.

Acronyms

DUT: Device **U**nder **T**est
TR: Testing **R**outer
G: Traffic **G**enerator

When several entities of the same type are present in a test configuration, a number is appended to the acronym to yield a label for each entity. For example, if there were three traffic generators in the test configuration, they would be labeled G1, G2 and G3.

Test Configuration



Basic Test Configuration

Traffic is passed from G1 to G2 via the DUT. The DUT may be configured as a router for some of the tests below which contain destination traffic to more than one network. When the term "Network Address" is used in the procedures below, it means that there is a range of IP addresses transmitted to a certain prefix that represents the destination subnet.

When a packet is denied by the DUT, there are 2 options. One is to send an error back to the

origin. This may be acceptable for some network configurations. The alternative is to silently discard the packet. While this is the more secure option, it may limit troubleshooting, especially if the network administrator has devices outside the firewall, such as that in a multi-site topology. In some cases, the testing may be run twice to observe the two alternate behaviors and verify they can be enabled and disabled.

TEST ORGANIZATION

This document organizes tests by group based on related test methodology or goals. Each group begins with a brief set of comments pertaining to all tests within that group. This is followed by a series of description blocks; each block describes a single test. The format of the description block is as follows:

Test Label:	The test label and title comprise the first line of the test block. The test label is composed by concatenating the short test suite name, the group number, and the test number within the group, separated by periods. The Test Number is the group and test number, also separated by a period. So, test label FIR.1.2 refers to the second test of the first test group in the Firewall test suite. The test number is 1.2.
Purpose:	The Purpose is a short statement describing what the test attempts to achieve. It is usually phrased as a simple assertion of the feature or capability to be tested.
References:	The References section lists cross-references to the specifications and documentation that might be helpful in understanding and evaluating the test and results.
Resource Requirements:	The Resource Requirements section specifies the software, hardware, and test equipment that will be needed to perform the test.
Discussion:	The Discussion is a general discussion of the test and relevant section of the specification, including any assumptions made in the design or implementation of the test as well as known limitations.
Test Setup:	The Test Setup section describes the configuration of all devices prior to the start of the test. Different parts of the procedure may involve configuration steps that deviate from what is given in the test setup. If a value is not provided for a protocol parameter, then the protocol's default is used for that parameter.
Procedure:	This section of the test description contains the step-by-step instructions for carrying out the test. These steps include such things as enabling interfaces, unplugging devices from the network, or sending packet from a test station. The test procedure also cues the tester to make observations, which are interpreted in accordance with the observable results given for that test part.
Observable Results:	This section lists observable results that can be examined by the tester to verify that the DUT is operating properly. When multiple observable results are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail for each test is usually based on how the DUT's behavior compares to the results described in this section.
Possible Problems:	This section contains a description of known issues with the test procedure, which may affect test results in certain situations.

REFERENCES

The following documents are referenced in this text:

[ADDRCONF] Thomson, S., T. Narten, IPv6 Stateless Address Autoconfiguration, RFC 2462, December 1998.

[ICMPv6] Conta, A., S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC 2463, December 1998.

[IPv6-SPEC] Hinden, R., S. Deering, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, December 1998.

[ND] Narten, T., Nordmark, E., and W. Simpson, Neighbor Discovery for IP Version 6 (IPv6), RFC 2461, December 1998.

[PMTU] McCann, J., S. Deering, and J. Mogul, Path MTU Discovery for IPv6, RFC 1981, August 1996.

[ADDR] Hinden, R., S. Deering, IP Version 6 Addressing Architecture, RFC 2373, July 1998.

[IP] Jon Postel, Editor. Internet Protocol: DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION, RFC 791, September 1981.

[TCP] Jon Postel, Editor. Internet Protocol: DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION, RFC 793, September 1981.

[UDP] Jon Postel. User Datagram Protocol, RFC 768, August 1980.

[RFC2827] P. Ferguson and D. Senie, Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827, May 2000.

[RFC3013] T. Killalea, Recommended Internet Service Provider Security Services and Procedures. RFC 3013, November 2000.

[FTP] Jon Postel. File Transfer Protocol (FTP), RFC 768, October 1985.

GROUP 1: Stateful Inspection

Scope:

These tests are designed to verify the functionality of the firewall inspection of connection state.

Overview:

Connection tracking between nodes across the firewall boundary is essential for high resolution security operations.

Test FIR.1.1: IPv6 TCP State

Purpose: To verify that a firewall properly accepts and denies IPv6 TCP traffic based on state.

References: TCP

Resource Requirements:

- Monitor to capture packets

Last Modification: February 20, 2004

Discussion: The TCP state machine (Annex B of [TCP]) can be observed by the Firewall. The firewall can block IPv6 TCP traffic that is received out of state.

Test Setup: Connect Devices as shown.

Procedure:

Part A: Proper Initialization Procedure

1. Configure the DUT to deny all TCP traffic.
2. From G1, transmit an IPv6 TCP SYN.
3. From G2, transmit an IPv6 TCP SYN-ACK.
4. From G1, transmit an IPv6 TCP ACK.
5. From G1 and G2, transmit traffic containing the proper source and destination TCP ports.
6. Observe the packets transmitted by the DUT.
7. Configure the DUT to allow stateful TCP connections to G1.
8. From G1, transmit an IPv6 TCP SYN.
9. From G2, transmit an IPv6 TCP SYN-ACK.
10. From G1, transmit an IPv6 TCP ACK.
11. From G1 and G2, transmit traffic containing the proper source and destination TCP ports.
12. Observe the packets transmitted by the DUT.

Part B: Reception of SYN-ACK before SYN

13. Configure the DUT to allow stateful TCP connections to G1.
14. From G1, transmit an IPv6 TCP SYN-ACK.
15. Observe the packets transmitted by the DUT.

Part C: Reception of ACK before SYN or SYN-ACK

16. Configure the DUT to allow stateful TCP connections to G1.
17. From G1, transmit an IPv6 TCP ACK.
18. Observe the packets transmitted by the DUT on G1 and G2.

Part D: TCP Cleanup Procedure

19. Configure the DUT to allow stateful TCP connections to G1.
20. From G1, transmit an IPv6 TCP SYN.
21. From G2, transmit an IPv6 TCP SYN-ACK.
22. From G1, transmit an IPv6 TCP ACK.
23. From G1 and G2, transmit traffic containing the proper source and destination TCP ports.
24. Observe the packets transmitted by the DUT on G1 and G2.

25. From G1, transmit an IPv6 TCP FIN.
26. From G2, transmit an IPv6 TCP FIN-ACK.
27. From G1, transmit an IPv6 TCP SYN-ACK.
28. Observe the packets transmitted by the DUT on G1 and G2.

Observable Results:

- *In Part A,*
  Step 6: The traffic transmitted from G1 must not be forwarded by the DUT.
  Step 12: The traffic transmitted from G1 must be forwarded by the DUT.
- *In Part B,*
  Step 15: The TCP SYN-ACK from G1 must not be forwarded by the DUT.
- *In Part C,*
  Step 18: The TCP ACK from G1 must not be forwarded by the DUT.
- *In Part D,*
  Step 24: The traffic transmitted from G1 must be forwarded by the DUT.
  Step 28: The TCP SYN-ACK from G1 must not be forwarded by the DUT.

Possible Problems:

- None.

Test FIR.1.2: FTP State Inspection

Purpose: To verify that a firewall properly accepts and denies FTP traffic based on state.

References: TCP, FTP

Resource Requirements:

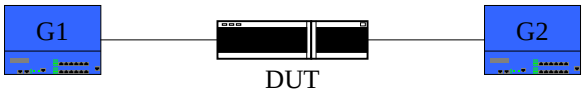
- Monitor to capture packets

Last Modification: November 18, 2004

Discussion: The TCP state machine (Annex B of [TCP]) can be observed by the Firewall. The firewall can block TCP traffic that is received out of state. The firewall must maintain separate TCP state for both IPv4 and IPv6.

Active FTP: client makes initial CMD connection to the server, then server makes DATA connection to client. (outgoing ftp connections are not normally blocked)
Passive FTP: client makes both CMD and DATA connections to the server. (incoming ftp data connections should be blocked)

Test Setup: Connect Devices as shown.



Procedure:

Part A: FTP connection in active mode

1. Configure the DUT to allow all traffic.
2. From G1, initialize an active FTP connection to G2 and transfer a file.
3. Observe the packets transmitted by the DUT.
4. Configure the DUT to deny all traffic.
5. From G1, initialize an active FTP connection to G2 and transfer a file.
6. Observe the packets transmitted by the DUT.
7. Configure the DUT to accept FTP traffic and deny all other traffic on the interface connected to G1.
8. From G1, initialize an active FTP connection to G2 and transfer a file.
9. Observe the packets transmitted by the DUT.

Part B: FTP connection in passive mode

10. Configure the DUT to allow all traffic.
11. From G1, initialize a passive FTP connection to G2 and transfer a file.
12. Observe the packets transmitted by the DUT.
13. Configure the DUT to deny all traffic.
14. From G1, initialize a passive FTP connection to G2 and transfer a file.
15. Observe the packets transmitted by the DUT.
16. Configure the DUT to accept FTP traffic and deny all other traffic on the interface connected to G1.
17. From G1, initialize a passive FTP connection to G2 and transfer a file.
18. Observe the packets transmitted by the DUT.

Observable Results:

- *In Part A,*
  Step 3: The file must be properly transferred from G1 to G2.
  Step 6: The file must not be transferred from G1 to G2.
  Step 9: The file must be properly transferred from G1 to G2.
- *In Part B,*
  Step 12: The file must be properly transferred from G1 to G2.
  Step 15: The file must not be transferred from G1 to G2.
  Step 18: The file should not be transferred from G1 to G2, due to data connection fail.

Possible Problems:

- Some devices might not support selection of active/passive ftp sessions.

.

Test FIR.1.3: ICMP State Inspection

Purpose: To verify that a firewall properly forwards ICMP reply messages based on state.

References: TCP, FTP

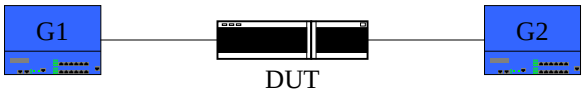
Resource Requirements:

- Monitor to capture packets

Last Modification: November 18, 2004

Discussion: ICMP alerts a host attempting to originate a connection that does not exist through a destination unreachable message. As fragmentation is done at the host in IPv6, it is necessary that hosts can receive Packet Too Big messages. Time Exceeded and Parameter Problem messages are also important error messages for a host to receive.

Test Setup: Connect Devices as shown.



Procedure:

Part A: ICMPv6 Destination Unreachable Message

1. Configure the DUT to allow all traffic.
2. From G1, transmit an ICMPv6 Destination Unreachable message.
3. Observe the packets transmitted by the DUT.
4. Configure the DUT to deny all traffic.
5. From G1, transmit an ICMPv6 Destination Unreachable message.
6. Observe the packets transmitted by the DUT.
7. Configure the DUT to deny messages that are out of state on the interface connected to G1.
8. From G1, transmit an ICMPv6 Destination Unreachable message.
9. Observe the packets transmitted by the DUT.
10. From G2, transmit an ICMPv6 Echo Request message to G1.
11. From G1, transmit an ICMPv6 Destination Unreachable message to G2.
12. Observe the packets transmitted by the DUT.

Part B: ICMPv6 Packet Too Big Message

13. Configure the DUT to allow all traffic.
14. From G1, transmit an ICMPv6 Packet Too Big message.
15. Observe the packets transmitted by the DUT.
16. Configure the DUT to deny all traffic.
17. From G1, transmit an ICMPv6 Packet Too Big message.
18. Observe the packets transmitted by the DUT.
19. Configure the DUT to deny messages that are out of state on the interface connected to G1.
20. From G1, transmit an ICMPv6 Packet Too Big message.

21. Observe the packets transmitted by the DUT.
22. From G2, transmit an ICMPv6 Echo Request message with a packet size of 1500 to G1.
23. From G1, transmit an ICMPv6 Packet Too Big message.
24. Observe the packets transmitted by the DUT.

Part C: ICMPv6 Time Exceeded Message

25. Configure the DUT to allow all traffic.
26. From G1, transmit an ICMPv6 Time Exceeded message.
27. Observe the packets transmitted by the DUT.
28. Configure the DUT to deny all traffic.
29. From G1, transmit an ICMPv6 Time Exceeded message.
30. Observe the packets transmitted by the DUT.
31. Configure the DUT to deny messages that are out of state on the interface connected to G1.
32. From G1, transmit an ICMPv6 Time Exceeded message.
33. Observe the packets transmitted by the DUT.
34. From G2, transmit an ICMPv6 Echo Request fragmented message to G1.
35. From G1, transmit an ICMPv6 Time Exceeded message.
36. Observe the packets transmitted by the DUT.

Part D: ICMPv6 Parameter Problem Message

37. Configure the DUT to allow all traffic.
38. From G1, transmit an ICMPv6 Parameter Problem message.
39. Observe the packets transmitted by the DUT.
40. Configure the DUT to deny all traffic.
41. From G1, transmit an ICMPv6 Parameter Problem message.
42. Observe the packets transmitted by the DUT.
43. Configure the DUT to deny messages that are out of state on the interface connected to G1.
44. From G1, transmit an ICMPv6 Parameter Problem message.
45. Observe the packets transmitted by the DUT.
46. From G2, transmit an ICMPv6 Echo Request message with an invalid payload length to G1.
47. From G1, transmit an ICMPv6 Parameter Problem message.
48. Observe the packets transmitted by the DUT.

Part E: ICMPv6 Echo Reply Message

49. Configure the DUT to allow all traffic.
50. From G1, transmit an ICMPv6 Echo Reply message.
51. Observe the packets transmitted by the DUT.
52. Configure the DUT to deny all traffic.
53. From G1, transmit an ICMPv6 Echo Reply message.
54. Observe the packets transmitted by the DUT.
55. Configure the DUT to monitor ICMPv6 traffic and deny messages that are out of state on the interface connected to G1.
56. From G1, transmit an ICMPv6 Echo Reply message.
57. Observe the packets transmitted by the DUT.
58. From G2, transmit an Echo Request to G1.
59. From G1, transmit an ICMPv6 Echo Reply message.
60. Allow the Observe the packets transmitted by the DUT on G1 and G2.

Observable Results:

- *In Part A,*

  Step 3: The Destination Unreachable message must be forwarded by the DUT.

Step 6: The Destination Unreachable message must not be forwarded by the DUT.
Step 9: The Destination Unreachable message must not be forwarded by the DUT.
Step 12: The Destination Unreachable message must be forwarded by the DUT.

- *In Part B,*
  Step 15: The Packet Too Big message must be forwarded by the DUT.
  Step 18: The Packet Too Big message must not be forwarded by the DUT.
  Step 21: The Packet Too Big message must not be forwarded by the DUT.
  Step 24: The Packet Too Big message must be forwarded by the DUT.
- *In Part C,*
  Step 27: The Time Exceeded message must be forwarded by the DUT.
  Step 30: The Time Exceeded message must not be forwarded by the DUT.
  Step 33: The Time Exceeded message must not be forwarded by the DUT.
  Step 36: The Time Exceeded message must be forwarded by the DUT.
- *In Part D,*
  Step 39: The Parameter Problem message must be forwarded by the DUT.
  Step 42: The Parameter Problem message must not be forwarded by the DUT.
  Step 45: The Parameter Problem message must not be forwarded by the DUT.
  Step 48: The Parameter Problem message must be forwarded by the DUT.
- *In Part E,*
  Step 51: The Echo Reply message must be forwarded by the DUT.
  Step 54: The Echo Reply message must not be forwarded by the DUT.
  Step 57: The Echo Reply message must not be forwarded by the DUT.
  Step 60: The Echo Reply message must be forwarded by the DUT.

Possible Problems:

- None.

Test FIR.1.4: UDP State Inspection

Purpose: To verify that a firewall properly forwards UDP DNS requests based on state.

References: TCP, FTP, RFC1035

Resource Requirements:

- Monitor to capture packets

Last Modification: February 28, 2004

Discussion: When a host makes a UDP DNS request, the DNS server must reply across the firewall. This test checks the functionality of this mechanism.

Test Setup: Connect Devices as shown. G1 is a DNS Server.

Procedure:

Part A: DNS Query

1. Configure the DUT to allow all traffic.
2. From G2, transmit a UDP DNS query message to G1.
3. Observe the packets transmitted by the DUT.
4. Configure the DUT to deny all traffic.
5. From G2, transmit a UDP DNS query message to G1.
6. Observe the packets transmitted by the DUT.
7. Configure the DUT to deny UDP DNS messages that are out of state on the interface connected to G1.
8. From G2, transmit a UDP DNS query message to G1.
9. From G1, transmit a UDP DNS reply message to G2.
10. Observe the packets transmitted by the DUT.

Part B: DNS Reply

11. Configure the DUT to allow all traffic.
12. From G1, transmit a UDP DNS reply message to G2.
13. Observe the packets transmitted by the DUT.
14. Configure the DUT to deny all traffic.
15. From G1, transmit a UDP DNS reply message to G2.
16. Observe the packets transmitted by the DUT.
17. Configure the DUT to deny UDP DNS messages that are out of state on the interface connected to G1.
18. From G1, transmit a UDP DNS reply message to G2.
19. Observe the packets transmitted by the DUT.
20. From G2, transmit a UDP DNS query message to G1.
21. From G1, transmit a UDP DNS reply message to G2.
22. Observe the packets transmitted by the DUT.

Part C: UDP State, destination UDP port.

23. Configure the DUT to allow all traffic.
24. From G2, transmit a UDP DNS request message to G1.
25. From G1, transmit a UDP DNS reply message to G2 with a UDP destination port of 7.
26. Observe the packets transmitted by the DUT.
27. Configure the DUT to deny all traffic.
28. From G2, transmit a UDP DNS request message to G1.
29. From G1, transmit a UDP DNS reply message to G2 with a UDP destination port of 7.
30. Observe the packets transmitted by the DUT.
31. Configure the DUT to deny UDP DNS messages that are out of state on the interface connected to G1.
32. From G2, transmit a UDP DNS request message to G1.
33. From G1, transmit a UDP DNS reply message to G2 with a UDP destination port of 7.
34. Observe the packets transmitted by the DUT.

Part D: UDP State, source UDP port.

35. Configure the DUT to allow all traffic.
36. From G2, transmit a UDP DNS request message to G1.
37. From G1, transmit a UDP DNS reply message to G2 with a UDP source port of 7.
38. Observe the packets transmitted by the DUT.
39. Configure the DUT to deny all traffic.
40. From G2, transmit a UDP DNS request message to G1.
41. From G1, transmit a UDP DNS reply message to G2 with a UDP source port of 7.
42. Observe the packets transmitted by the DUT.
43. Configure the DUT to deny UDP DNS messages that are out of state on the interface connected to G1.
44. From G2, transmit a UDP DNS request message to G1.
45. From G1, transmit a UDP DNS reply message to G2 with a UDP source port of 7.
46. Observe the packets transmitted by the DUT.

Part E: UDP State, source IP address.

47. Configure the DUT to allow all traffic.
48. From G2, transmit a UDP DNS request message to G1.
49. From G1, transmit a UDP DNS reply message to G2 with a source address that is not the destination address of the DNS request.
50. Observe the packets transmitted by the DUT.
51. Configure the DUT to deny all traffic.
52. From G2, transmit a UDP DNS request message to G1.
53. From G1, transmit a UDP DNS reply message to G2 with a source address that is not the destination address of the DNS request.
54. Observe the packets transmitted by the DUT.
55. Configure the DUT to deny UDP DNS messages that are out of state on the interface connected to G1.
56. From G2, transmit a UDP DNS request message to G1.
57. From G1, transmit a UDP DNS reply message to G2 with a source address that is not the destination address of the DNS request.
58. Observe the packets transmitted by the DUT.

Part F: UDP State, destination IP address

59. Configure the DUT to allow all traffic.
60. From G2, transmit a UDP DNS request message to G1.
61. From G1, transmit a UDP DNS reply message to G2 with a destination address that is not the source address of the DNS request.
62. Observe the packets transmitted by the DUT.

63. Configure the DUT to deny all traffic.
64. From G2, transmit a UDP DNS request message to G1.
65. From G1, transmit a UDP DNS reply message to G2 with a destination address that is not the source address of the DNS request.
66. Observe the packets transmitted by the DUT.
67. Configure the DUT to deny UDP DNS messages that are out of state on the interface connected to G1.
68. From G2, transmit a UDP DNS request message to G1.
69. From G1, transmit a UDP DNS reply message to G2 with a destination address that is not the source address of the DNS request.
70. Observe the packets transmitted by the DUT.

Observable Results:

- *In Part A,*
  Step 3: The DNS query message must be forwarded by the DUT.
  Step 6: The DNS query message must not be forwarded by the DUT.
  Step 10: The DNS query message must be forwarded by the DUT.
- *In Part B,*
  Step 13: The DNS reply message must be forwarded by the DUT.
  Step 16: The DNS reply message must not be forwarded by the DUT.
  Step 19: The DNS reply message must not be forwarded by the DUT.
  Step 22: The DNS reply message must be forward by the DUT.
- *In Part C,*
  Step 26: The DNS reply message must be forwarded by the DUT.
  Step 30: The DNS reply message must not be forwarded by the DUT.
  Step 34: The DNS reply message must not be forwarded by the DUT.
- *In Part D,*
  Step 38: The DNS reply message must be forwarded by the DUT.
  Step 42: The DNS reply message must not be forwarded by the DUT.
  Step 46: The DNS reply message must not be forwarded by the DUT.
- *In Part E,*
  Step 50: The DNS reply message must be forwarded by the DUT.
  Step 54: The DNS reply message must not be forwarded by the DUT.
  Step 58: The DNS reply message must not be forwarded by the DUT.
- *In Part F,*
  Step 62: The DNS reply message must be forwarded by the DUT.
  Step 66: The DNS reply message must not be forwarded by the DUT.
  Step 70: The DNS reply message must not be forwarded by the DUT.

Possible Problems:

- None.

GROUP 2: Advanced Filtering

Scope:

These tests are designed to verify the functionality of the firewall in a multiprotocol environment.

Overview:

Connection tracking between nodes across the firewall boundary is essential for high resolution security. Simulating realistic operations is necessary to create a better indicator of the firewall operation.

Test FIR.2.1: Multiprotocol Filtering

Purpose: To verify that a firewall properly accepts and denies various application traffic based on state in the presence of mixed IPv4 and IPv6 traffic.

References: TCP, UDP

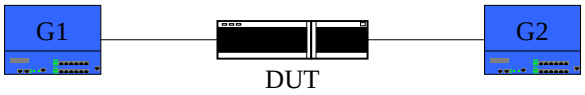
Resource Requirements:

- Monitor to capture packets

Last Modification: November 18, 2004

Discussion: To test firewall filters in a realistic setting (various UDP and TCP ports and IP addresses) application traffic is sent that "cycles" through, or randomizes, addresses/ports from address ranges. This tests the firewall's ability to block the "bad" (filtered) traffic whilst passing all the "good" traffic, using a very large range of addresses/ports, over a period of time.

Test Setup: Connect Devices as shown.



Procedure:

Part A: Filter of Application Traffic

1. Configure the DUT to accept only IPv6 application traffic from HTTP and TELNET traffic.
2. From G1, transmit IPv6 TELNET traffic to G2.
3. From G1, transmit IPv6 FTP traffic to G2.
4. From G1, transmit IPv6 HTTP traffic to G2.
5. Observe the packets transmitted by the DUT.

Part B: Dynamic Firewall Configuration

6. Configure the DUT to accept only IPv6 application traffic from HTTP and TCP.
7. From G1, transmit IPv6 TCP traffic to G2.
8. From G1, transmit IPv6 FTP traffic to G2.
9. From G1, transmit IPv6 HTTP traffic to G2.
10. Observe the packets transmitted by the DUT
11. Configure the DUT to accept only IPv6 application traffic from HTTP.
12. From G1, transmit IPv6 TCP traffic to G2.
13. From G1, transmit IPv6 FTP traffic to G2.
14. From G1, transmit IPv6 HTTP traffic to G2.
15. Observe the packets transmitted by the DUT.

Observable Results:

- *In Part A,*
  Step 5: The DUT must forward the HTTP and TELNET traffic. The FTP traffic must not be forwarded by the DUT.
- *In Part B,*

Step 10: The DUT must forward the HTTP and TELNET traffic. The FTP traffic must not be forwarded by the DUT.
Step 15: The DUT must forward the HTTP traffic. The TELNET and FTP traffic must not be forwarded by the DUT.

Possible Problems:

- None.

Test FIR.2.2: Firewall Screening

Purpose: To verify that a firewall properly defends against potential Denial of Service attacks.

References: IPv6-SPEC
RFC2827
RFC3013

Resource Requirements:

- Monitor to capture packets

Last Modification: September 9, 2004

Discussion: Denial of Service attacks send disruptive traffic to hosts on a network. This traffic slows or stalls access to those hosts and/or damages the hosts by flooding their buffers and exploiting operating system security holes to gain access. Common attacks include TCP SYN flood, UDP flood, ICMP flood, Spoofing. When behaving appropriately, the firewall should allow the packets to go through when they are sent at a low rate and stop them when being sent at higher than expected rate. The values defined as rates are predetermined values dependent on the DUT.

Test Setup: Connect Devices as shown.

Procedure:

Part A: TCP SYN Flood

1. Configure the DUT to prevent a TCP SYN flood situation on the interface connected to G1.
2. From G1 to G2, transmit TCP packets with the SYN bit set using variable IPv6 source addresses at a lower rate than configured in step 1.
3. Observe the packets transmitted by the DUT.
4. From G1, transmit TCP traffic to G2 with the SYN bit set using variable IPv6 source addresses. This traffic should exceed the rate configured in Step 1.
5. Observe the packets transmitted by the DUT.

Part B: UDP Flood

6. Configure the DUT to prevent a UDP flood situation on the interface connected to G1.
7. From G1 to G2, transmit UDP packets using variable IPv6 source addresses at a lower rate than configured in step 6.
8. Observe the packets transmitted by the DUT.
9. From G1, transmit UDP traffic to G2 using variable IPv6 source addresses. This traffic should exceed the rate configured in Step 6.
10. Observe the packets transmitted by the DUT.

Part C: ICMP Flood

11. Configure the DUT to prevent an ICMPv6 flood situation on the interface connected to G1.

12. From G1 to G2, transmit ICMPv6 Echo Request messages using variable IPv6 source addresses at a lower rate than configured in step 11.
13. Observe the packets transmitted by the DUT.
14. From G1, transmit ICMPv6 Echo Request messages to G2 with the SYN bit set using variable IPv6 source addresses. This traffic should exceed the rate configured in Step 11.
15. Observe the packets transmitted by the DUT.

Part D: Simultaneous DoS attacks with Application Data

16. Configure the DUT to prevent an ICMPv6, TCP SYN, and UDP flood situations on the interface connected to G1.
17. From G1 to G2, transmit ICMPv6 Echo Request messages using variable IPv6 source addresses at a lower rate than configured in step 13.
18. Observe the packets transmitted by the DUT.
19. From G1, transmit TCP traffic to G2 with the SYN bit set using variable IPv6 source addresses. This traffic should exceed the rate configured in step 13.
20. Observe the packets transmitted by the DUT.

Observable Results:

- *In Part A,*
  Step 3: The packets should be forwarded by the DUT.
  Step 5: The packets exceeding the allowed rate should be dropped.
- *In Part B,*
  Step 8: The packets should be forwarded by the DUT.
  Step 10: The packets exceeding the allowed rate should be dropped.
- *In Part C,*
  Step 12: The packets should be forwarded by the DUT.
  Step 15: The packets exceeding the allowed rate should be dropped.
- In Part D,
  Step 18: The ICMPv6 packets should be forwarded by the DUT.
  Step 20: The TCP packets exceeding the allowed rate should be dropped.

Possible Problems:

- The DUT might not support filtering by rate.

GROUP 3: Firewall Performance

Scope:

These tests are designed to evaluate the IPv6 performance capabilities of the firewall device.

Overview:

It is essential to understand the performance of a firewall, such as maximum application throughput, total TCPv6 concurrent connections and TCPv6 connection setup rate.

Test FIR.3.1: Maximum Concurrent TCPv6 Connections

Purpose: To determine the maximum number of concurrent TCPv6 connections supported through or with the DUT.

References:

Resource Requirements:

Last Modification: October 29, 2004

Discussion: Firewalls keep track of the TCPv6 connections passing through it in a state table. This test will assist in verifying how well this state table is able to scale.

Test Setup: Connect Devices as shown.

Procedure:

1. On G1, configure emulated client(s) to initiate an HTTPv6 connection to an application server(s) emulated on G2.
2. Configure the following test parameters:
   - Desired number of concurrent TCPv6 connections
   - Number of GET requests per TCPv6 connection (default = 1)
   - HTTP requested file (default = 16 bytes)
   - Delay between GET requests (default = 1)
   - TCPv6 connection setup rate (default = 100)
3. Configure the test to keep all TCPv6 connections open for the duration of the test.
4. From G1, start initiating HTTPv6 connections to G2.
5. Use an iterative search algorithm, where for each iteration the desired number of concurrent TCPv6 connections attempted is increased. Continue test iterations until one or more connection attempts fails.

Variables:

The following test variables can be modified to find different test outcomes:

1. Number of GET requests per TCPv6 connection
2. HTTP requested file
3. Delay between GET requests per TCPv6 connection
4. TCPv6 connection setup rate

Observable Results:

1. Maximum concurrent TCPv6 connections (the value before a connection attempt has failed).
2. Total HTTP GET requests completed.

Possible Problems:

- None.

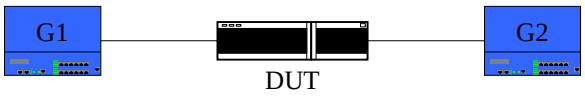
Test FIR.5.2: Maximum TCPv6 Connection Setup Rate

Purpose: To determine the maximum TCPv6 connection establishment rate through or with the DUT.

Last Modification: October 29, 2004

Discussion: Firewalls keep track of the TCPv6 connections passing through it in a state table. This test will assist in verifying how well the DUT can update its state table.

Test Setup: Connect Devices as shown.



Procedure:

1. On G1, configure emulated client(s) to initiate an HTTPv6 connection to an application server(s) emulated on G2.
2. Configure the following test parameters:
    a. Initial TCPv6 connection setup rate (default = 100)
    b. Final (maximum) TCPv6 connection setup rate (default = 1000)
    c. Elapsed time between initial and final TCPv6 connection setup rate (default = 60sec)
    d. Number of GET requests per TCPv6 connection (default = 1)
    e. HTTP requested file (default = 16 bytes)
    f. Delay between GET requests (default = 1)
3. From G1, start initiating HTTPv6 connections to G2.
4. Observe test results as TCPv6 connection setup rate is iteratively increased from the initial to the final connection rate. The final connection rate where no connection attempts failed will be the maximum TCPv6 connection rate. Repeat the test until the optimal connection ramp rate has been achieved.

Variables:

The following test variables can be modified to find different test outcomes:

1. Number of GET requests per TCPv6 connection
2. HTTP requested file
3. Delay between GET requests per TCPv6 connection

Observable Results:

1. Maximum TCPv6 connection setup rate (the value before a connection attempt has failed).
2. Total HTTP GET requests completed.

Possible Problems:

- None.

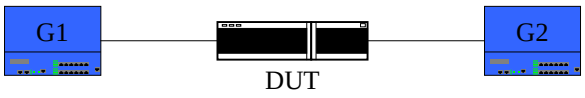
Test FIR.3.3: Maximum TCPv6 Connection Teardown Rate

Purpose: To determine the maximum TCPv6 connection teardown rate through or with the DUT.

Last Modification: October 29, 2004

Discussion: Firewalls keep track of the TCPv6 connections passing through it in a state table. This test will assist in verifying how well the DUT is able to release TCPv6 connections it is keeping track of.

Test Setup: Connect Devices as shown.



Procedure:

1. On G1, configure emulated client(s) to initiate an HTTPv6 connection to an application server(s) emulated on G2.
2. Configure the following test parameters:
    a. Number of concurrent TCPv6 connections
    b. Number of HTTP GET requests per TCPv6 connection (default = 1)
    c. HTTP requested file (default = 16 bytes)
    d. Delay between GET requests (default = 1)
3. Configure the test to keep all TCPv6 connections open for a specified hold time.
4. Configure the test to teardown all TCPv6 connections over a teardown duration.
5. From G1, start initiating HTTPv6 connections to G2 and open desired number of concurrent TCPv6 connections.
6. Observe test results as TCPv6 connections are iteratively torn down after the connection hold time expires. Repeat the test until the maximum teardown rate (modify teardown duration) has been achieved.

Variables:

The following test variables can be modified to find different test outcomes:

1. Number of GET requests per TCPv6 connection
2. HTTP requested file
3. Delay between GET requests per TCPv6 connection
4. Number of concurrent TCPv6 connections

Observable Results:

1. Maximum TCPv6 connection teardown rate (the rate where all TCPv6 connections have successfully been closed).

Possible Problems:

- None.

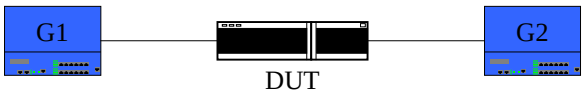
Test FIR.3.4: Maximum Application Transaction Rate

Purpose: Determine the maximum application transaction rate the DUT is able to sustain.

Last Modification: October 29, 2004

Discussion: Firewalls keep track of the TCPv6 connections passing through it in a state table. Firewalls are also able to inspect traffic through to the application layer. This test will verify how well the DUT can sustain application performance with and without application inspection.

Test Setup: Connect Devices as shown.



Procedure:

Part A: HTTP Transaction Rate

2. On G1, configure emulated client(s) to initiate an HTTPv6 connection to an application server(s) emulated on G2.
3. Configure the following test parameters:
    a. Initial HTTP transaction rate (default = 1000)
    b. Final (maximum) HTTP transaction rate (default = 10000)
    c. Elapsed time between initial and final HTTP transaction rate (default = 60sec)
    d. Number of GET requests per TCPv6 connection (default = 10000)
    e. HTTP requested file (default = 16 bytes)
    f. Delay between GET requests (default = 0)
4. From G1, start initiating HTTPv6 connections to G2.
5. Observe test results as the HTTP transaction rate is iteratively increased from the initial to the final transaction rate. The final transaction rate where no transaction attempts failed will be the maximum HTTP transaction rate. Repeat the test until the optimal transaction ramp rate has been achieved.

Part B: HTTP Transaction Rate with Application Inspection

6. If available, configure DUT to enable any HTTP inspection/filtering capabilities.
7. Repeat test as outlined in Part A.
8. Observe the impact of application inspection on the baseline HTTP transaction rate measured in Part A.

Part C: FTP Transaction Rate

9. Repeat test as outlined in Part A, however, using FTPv6 active connections and FTP GET requests.

Part D: FTP Transaction Rate with Application Inspection

10. If available, configure DUT to enable any FTP inspection/filtering capabilities.
11. Repeat test as outlined in Part C.
12. Observe the impact of application inspection on the baseline FTP transaction rate measured in Part C.

Part E: Mixed HTTP/FTP Transaction Rate

13. Run tests outlined Parts A and C simultaneously.
14. Observe the impact of mixed application traffic on the baseline HTTP and FTP transaction rates measured in Part A and C.

Part F: Mixed HTTP/FTP Transaction Rate with Application Inspection

15. Run tests outlined Parts B and D simultaneously.
16. Observe the impact of application inspection on the baseline HTTP and FTP transaction rate measured in Parts B and D.

Variables:

The following test variables can be modified to find different test outcomes:

1. Number of GET requests per TCPv6 connection
2. HTTP requested file
3. Delay between GET requests per TCPv6 connection

Observable Results:

1. Maximum HTTP transaction rate
2. Total HTTP GET requests completed
3. Maximum FTP transaction rate
4. Total FTP GET requests completed

Possible Problems:

- Application inspection is not available on the DUT.

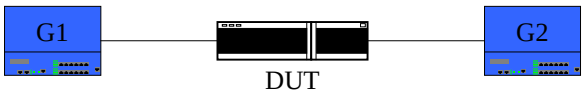
Test FIR.3.5: Voice and Data Application Traffic

Purpose: Verify the capability of the DUT to support both data and voice traffic through it.

Last Modification: October 29, 2004

Discussion: Firewalls keep track of the TCPv6 connections passing through it in a state table. Firewalls are also able to inspect traffic through to the application layer. This test will verify the ability of the DUT to support both voice and data connections through it.

Test Setup: Connect Devices as shown.



Procedure:

Part A: SIP Call

1. On G1 and G2, configure emulated SIP endpoint(s).
2. Configure the following test parameters:
    a. Call duration (default = 10sec)
    b. Audio file to play
    c. Number of calls (default = 1)
3. From endpoint emulated on G1, initiate SIP call to endpoint emulated on G2.

Part B: H323 Call

4. On G1 and G2, configure emulated H323 endpoint(s).
5. Configure the following test parameters:
    a. Call duration (default = 10sec)
    b. Audio file to play
    c. Number of calls (default = 1)
6. From endpoint emulated on G1, initiate H323 call to endpoint emulated on G2.

Part C: SIP/H323 Calls

7. Run tests outlined in Parts A and B simultaneously.

Part D: Data and Voice Traffic

8. On G1, configure emulated client(s) to initiate an HTTPv6 connection to an application server(s) emulated on G2.
9. On G1 and G2, configure emulated SIP and H323 endpoint(s).
10. Start generating a sustained rate of HTTPv6 transactions through the DUT.
11. From endpoints emulated on G1, initiate SIP and H323 calls to endpoints emulated on G2.

Variables:

The following test variables can be modified to find different test outcomes:

1. Number of emulated voice endpoints
2. Voice call duration
3. Number of calls to initiate

Observable Results:

1. In Part A, observe the SIP voice call has been successfully established, played and torn down through the DUT.
2. In Part B, observe the H323 voice call has been successfully established, played and torn down through the DUT.
3. In Part C, observe both SIP and H323 voice calls have been successfully established, played and torn down through the DUT.
4. In Part D, observe both SIP and H323 voice calls have been successfully established, played and torn down through the DUT, while the DUT is processing/inspecting HTTP traffic.

Possible Problems:

- SIP may not be supported by DUT.

GROUP 4: IPsecv6

Scope:

These tests are designed to verify the support for IPsecv6 on a firewall device.

Overview:

IPsec is a mandatory component of IPv6 implementations, so it is essential for a firewall to process IPsec-secured packets, as well establish IPsec connections (tunnels).

Test FIR.4.1: IPsecv6 Tunneling

Purpose: Verify DUT is able to support being an endpoint of an Ipsecv6 tunnel.

References:

Last Modification: October 29, 2004

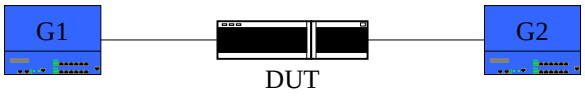
Discussion: Firewalls are able to process and identify IPsec-secured packets.
The use of HMAC-SHA-1-96 algorithm [RFC-2404] within AH and ESP MUST be supported.
The use of HMAC-MD5-96 algorithm [RFC-2403] within AH and ESP MAY also be supported.

Since ESP encryption and authentication are both optional, support for the NULL encryption algorithm [RFC-2410] and the NULL authentication algorithm [RFC-2406] MUST be provided to maintain consistency with the way these services are negotiated. However, while authentication and encryption can each be NULL, they MUST NOT both be NULL.

The *3DES-CBC* encryption algorithm [RFC-2451] does not suffer from the same security issues as DES-CBC, and the 3DES-CBC algorithm within ESP MUST be supported to ensure interoperability.

The *AES-128-CBC* algorithm [RFC-3602] MUST also be supported within ESP. AES-128 is expected to be a widely available, secure, and efficient algorithm. While AES-128-CBC is not required by thecurrent IPsec RFCs, it is expected to become required in the future.

Test Setup: Connect Devices as shown. Encryptions types should be 3DES-CBC, AES-128-CBC, and NULL.



Procedure:

Part A:

1. Configure G1 to transmit valid encrypted AH only packets carrying an http session with G2.
2. Using the same static key static key configure DUT to decrypt *for AH* traffic destined for G1.

Part B:

3. Configure G1 to transmit valid encrypted ESP only packets carrying an http session with G2.
4. Using the same static key static key configure DUT to decrypt *for ESP* traffic destined for G1.

Part C:

5. Configure G1 to transmit valid encrypted AH and ESP only packets carrying an http session with G2.
6. Using the same static key static key configure DUT to decrypt *for AH and ESP* traffic destined for G1.

Observable Results:

- In all parts, verify that the DUT is properly forwarding the IPsecv6 traffic coming from G1, to G2 un-encrypted. Verify that the DUT is properly forwarding the IPv6 traffic coming from G2, encrypted via Ipsecv6 to G1.

Possible Problems:

- None.

Test FIR.4.2: IPsecv6 Tunnel Establishment

Purpose: Verify DUT is able to establish an IPsecv6 connection against a peering device.

References:

Last Modification: October 29, 2004

Discussion: Firewalls are able to process and identify IPsec-secured packets as well as respond to IPsec connection requests.

Test Setup: Connect Devices as shown.

Procedure:

1. On G1, configure emulated IPsec client(s) to initiate an IPsecv6 connection to the DUT.
2. Configure the following Phase I IPsec connection parameters on G1 and the DUT:
    a. Authentication algorithm (default = MD5)
    b. Encryption algorithm (default = 3DES)
    c. DH group (Default = 2)
    d. Pre-shared key authentication method
3. Configure the following Phase II IPsec connection parameters on G1 and the DUT:
    a. Authentication algorithm (default = MD5)
    b. Encryption algorithm (default = 3DES)
    c. IPsec protocol (Default = ESP)
    d. IPsec encapsulation (Default = Tunnel)
4. From emulated IPsec clients on G1, initiate IPsec connections to the DUT.
5. Repeat test with different Phase I and Phase II parameter settings.

Variables:

The following test variables can be modified to find different test outcomes:

1. Number of IPsec tunnels
2. Authentication algorithm
3. Encryption algorithm
4. DH group
5. IPsec protocol
6. IPsec encapsulation

Observable Results:

- In Part A, verify that IPsecv6 tunnels are successfully established.

Possible Problems:

- None.

Test FIR.4.3: Traffic over IPsecv6 Tunnel

Purpose: Verify DUT is able to process incoming/outgoing traffic transported over an IPsecv6 tunnel.

References:

Resource Requirements:

Last Modification: October 29, 2004

Discussion: Firewalls are able to process and identify IPsec-secured packets as well as respond to IPsec connection requests.

Test Setup: Connect Devices as shown.

Procedure:

1. On G1, configure emulated IPsecv6 client(s) to initiate an IPsecv6 connection to the DUT.
2. On G1, configure emulated IPsecv6 client(s) to initiate an HTTPv6 connection to an application server(s) emulated on G2.
3. Configure the following Phase I IPsec connection parameters on G1 and the DUT:
    a. Authentication algorithm (default = MD5)
    b. Encryption algorithm (default = 3DES)
    c. DH group (Default = 2)
    d. Pre-shared key authentication method
4. Configure the following Phase II IPsec connection parameters on G1 and the DUT:
    a. Authentication algorithm (default = MD5)
    b. Encryption algorithm (default = 3DES)
    c. IPsec protocol (Default = ESP)
    d. IPsec encapsulation (Default = Tunnel)
5. From emulated IPsec clients on G1, initiate IPsecv6 connections to the DUT.
6. From emulated IPsec clients on G1, initiate HTTPv6 connections to G2.
7. Repeat test with different Phase I and Phase II parameter settings.

Variables:

The following test variables can be modified to find different test outcomes:

1. Number of IPsec tunnels
2. Authentication algorithm
3. Encryption algorithm
4. DH group
5. IPsec protocol
6. IPsec encapsulation
7. Number of HTTP GET requests per TCPv6 connection
8. HTTP requested file

Observable Results:

- In Part A, verify that IPsecv6 tunnels are successfully established, and secured HTTP transactions are successfully completed.

Possible Problems:

- None.

Test FIR.4.4: Mixed IPv6 and IPsecv6 traffic

Purpose: Verify DUT is able to process secured IPv6 traffic as well as unsecured IPv6 traffic.

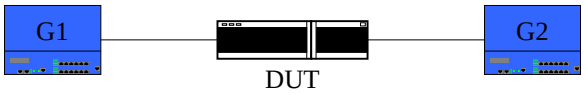
References:

Resource Requirements:

Last Modification: October 8, 2004

Discussion: Firewalls are able to process and identify IPsec-secured packets as well as respond to IPsec connection requests.

Test Setup: Connect Devices as shown.



Procedure:

1. On G1, configure emulated IPsecv6 client(s) to initiate an IPsecv6 connection to the DUT.
2. On G1, configure emulated IPsecv6 client(s) to initiate an HTTPv6 connection to an application server(s) emulated on G2.
3. On G1, configure non-secured emulated client(s) to initiate an HTTPv6 connection to an application server(s) emulated on G2.
4. Configure the following Phase I IPsec connection parameters on G1 and the DUT:
    a. Authentication algorithm (default = MD5)
    b. Encryption algorithm (default = 3DES)
    c. DH group (Default = 2)
    d. Pre-shared key authentication method
5. Configure the following Phase II IPsec connection parameters on G1 and the DUT:
    a. Authentication algorithm (default = MD5)
    b. Encryption algorithm (default = 3DES)
    c. IPsec protocol (Default = ESP)
    d. IPsec encapsulation (Default = Tunnel)
6. From emulated IPsec clients on G1, initiate IPsecv6 connections to the DUT.
7. From emulated IPsec clients on G1, initiate HTTPv6 connections to G2.
8. From non-secured clients on G1, initiate HTTPv6 connections to G2.
9. Repeat test with different Phase I and Phase II parameter settings

Variables:

The following test variables can be modified to find different test outcomes:

1. Number of IPsec tunnels
2. Authentication algorithm
3. Encryption algorithm
4. DH group
5. IPsec protocol
6. IPsec encapsulation

7. Number of HTTP GET requests per TCPv6 connection
8. HTTP requested file

Observable Results:

- In Part A, verify that IPsecv6 tunnels are successfully established, and secured HTTP transactions are successfully completed. Non-secured HTTP transactions should also be successfully completed.

Possible Problems:

- None.