

Moonv6 Test Suite
IPv6 Access Policy
Functionality Test Suite

Technical Document

Revision 1.0



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

ACKNOWLEDGEMENTS	4
INTRODUCTION	5
TEST ORGANIZATION.....	7
REFERENCES	8
GROUP 1: Basic Access Policy.....	9
Test AP.1.1: IPv4 Authorization	10
Test AP.1.2: Source Address Denial.....	12
Test AP.1.3: Destination Address Denial.....	14
Test AP.1.4: UDP Port Numbers	16
Test AP.1.5: TCP Port Numbers	18
Test AP.1.6: ICMPv6 Traffic.....	20
Test AP.1.7: ICMPv6 Control Traffic for Mobile IPv6.....	23
Test AP.1.8: Hop-by-Hop Header.....	25
Test AP.1.9: Default Router.....	26
Test AP.1.10: Time Based Authorization	28
Test AP.1.11: IPSec Forwarding.....	30
Test AP.1.12: MAC Authorization	31
Test AP.1.13: ICMPv6 Destination Unreachable	33
GROUP 2: Advanced AP Functionality	34
Test AP.2.1: Combination Authorization.....	35
Test AP.2.2: Ordered List Policy	37
Test AP.2.3: IPv4 and IPv6 Functionality, Same Policy.....	38
Test AP.2.4: IPv4 and IPv6 Functionality, Different Policy.....	40
Test AP.2.5: Tiny Fragment Attack for IPv4 and IPv6.....	42
Test AP.2.6: Overlapping Fragment Attack for IPv4 and IPv6	44
Test AP.2.7: Route Distribution.....	46
Test AP.2.8: Long Access Policy List Forwarding	48

MODIFICATION RECORD

Draft Version Complete August 11, 2004

Version 0.3 August 15, 2004

Version 0.5 September 9, 2004

Created a new set of test plans including Policy, Base Firewall and Firewall Interoperability.

Version 1.0 October 31, 2004

Scanned for errors and finished discussions and references. Approved for release.

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, Cisco and NetScreen for contributing test ideas for the base document of the Moonv6 Firewall Functionality test plan, from which this document is based.

Yoni Appel	Check Point Software Technologies
Ankur Chadda	University of New Hampshire
Eli Ginot	Check Point Software Technologies
Paul Meyer	Secure Computing
Jeff Pomeroy	Secure Computing
Kari Revier	University of New Hampshire
Benjamin Schultz	University of New Hampshire
Shinsuke Suzuki	Hitachi Ltd.
L. Brad Upson	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 the policy functionality of router products. The tests do not determine if a product conforms to any 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 any other 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 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 equipment 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

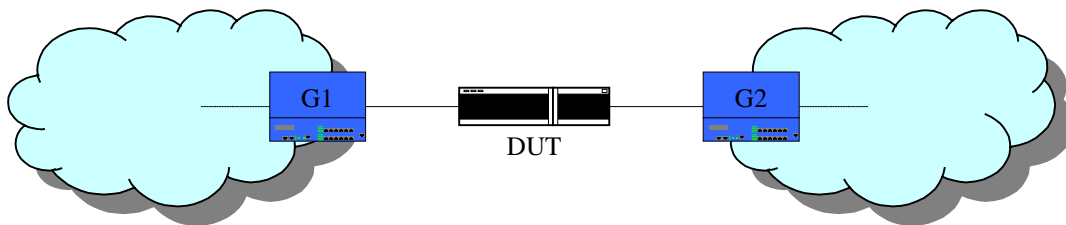
DUT: Device Under Test

TR: Testing Router

G: Traffic Generator

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.

Table of Testing Combinations

Logging Status Device Mode	Logging Disabled	Log only Denial Attempts	Log Accepts and Denies
Static IP Filter	Static IP Filter, Disable Log	Static IP Filter, Log Denials	Static IP Filter, Log Accepts and Denies
Proxy	Proxy, Disable Log	Proxy, Log Denials	Proxy, Log Accepts and Denies
Stateful Firewall	Stateful Firewall, Disable Log	Stateful Firewall, Log Denials	Proxy, Log Accepts and Denies

Logging functionality is a key feature that is tested with this test plan. This test plan can be run with logging disabled. It can also be run with logging of denials and logging of both allows and denies.

All tests in this test plan can be run in three different modes, depending on what the DUT supports. If the device is a traditional IP router, the tests below can be run in IP Filter mode. If the device is a proxy, the tests can be run in proxy mode. If the device is a stateful firewall, the tests can be run in stateful inspection mode.

This test suite can be run in nine different combinations as indicated in the table above.

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 AP.1.2 refers to the second test of the first test group in the AP 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.
Last Modification	The last date this test was modified.
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.

- [RFC3775] D. Johnson, C. Perkins and J. Arkko, Mobility Support in IPv6. RFC 3775, June, 2004.

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

- [RFC1858] G. Ziemba, D. Reed and P. Traina. Security Considerations for IP Fragment Filtering, RFC 1858, October 1995.

- [IEEE802] IEEE Std 802-1990. IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture. December, 1990.

GROUP 1: Basic Access Policy

Scope:

These following tests are designed to verify basic functionality and operation of IPv6-based access policy.

Overview:

IP Routers are designed to forward traffic between IP subnets. Access Policies (APs) are static accept and deny features that some IP routers have to limit access between networks. This reduces the ability to attack insecure hosts and the information on them. Acceptance and rejection policy can be based upon IP address, time and/or TCP/UDP fields.

Test AP.1.1: IPv4 Authorization

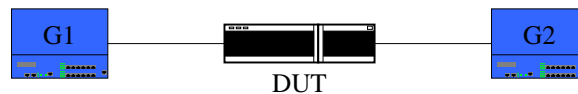
Purpose: To verify that a device has basic IPv4 access policy functionality.

References: IP

Resource Requirements: Monitor to capture packets, generators

Discussion: Basic IPv4 functionality includes the ability to accept and deny source and destination address pairs on an IPv4 network. This test can be repeated with unicast, broadcast and multicast addresses.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny traffic with an IPv4 Unicast Destination Address

1. Configure the DUT to deny traffic from a range of destination IPv4 unicast addresses on the interface connected to G1.
2. From G1, transmit traffic to G2 containing the destination IPv4 unicast address range configured in the previous step.
3. From G1, transmit traffic to G2 containing a valid destination address range.
4. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic with an Unicast IPv4 Source Address

5. Configure the DUT to deny traffic from a range of source IPv4 unicast addresses on the interface connected to G1.
6. From G1, transmit traffic to G2 containing the source IPv4 unicast address range configured in the previous step.
7. From G1, transmit traffic to G2 containing a valid source address range.
8. Observe the packets transmitted by the DUT on G2.

Part C: Deny traffic with an IPv4 Multicast Destination Address

9. Configure the DUT to deny traffic from a range of destination IPv4 multicast addresses on the interface connected to G1.
10. From G1, transmit traffic to G2 containing the destination IPv4 multicast address range configured in the previous step.
11. From G1, transmit traffic to G2 containing a valid destination multicast address range.
12. Observe the packets transmitted by the DUT on G2.

Part D: Deny traffic with an IPv4 Multicast Source Address

13. Configure the DUT to deny traffic from a range of source IPv4 multicast addresses on the interface connected to G1.
14. From G1, transmit traffic to G2 containing the source IPv4 multicast address range configured in the previous step.
15. From G1, transmit traffic to G2 containing a valid source address range.
16. Observe the packets transmitted by the DUT on G2.

Part E: Deny traffic with an IPv4 Broadcast Destination Address

17. Configure the DUT to deny traffic from the IPv4 broadcast destination address on the interface connected to G1.
18. From G1, transmit traffic to G2 containing the IPv4 broadcast address.

19. Observe the packets transmitted by the DUT on G2.

Part F: Accept all traffic with an IPv4 Broadcast Destination Address

20. Configure the DUT to accept all traffic from the IPv4 broadcast destination address on the interface connected to G1.

21. From G1, transmit traffic to G2 containing the IPv4 broadcast address.

22. Observe the packets transmitted by the DUT on G2.

Part G: Deny all traffic with an IPv4 Broadcast Source Address

23. Configure the DUT to deny all traffic from an IPv4 Broadcast Source Address on the interface connected to G1.

24. From G1, transmit traffic to G2 containing an IPv4 Broadcast Source Address.

25. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Parts A and C, the traffic transmitted from denied destination address range on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the valid addresses at G1 should be forwarded by the DUT and observed on G2.
- In Parts B and D, the traffic transmitted from denied source address range on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the valid addresses at G1 should be forwarded by the DUT and observed on G2.
- In Part E, the traffic transmitted from the broadcast IPv4 destination address on G1 should not be forwarded by the DUT and not be observed on G2.
- In Part F, the traffic transmitted from the broadcast IPv4 destination address on G1 should be forwarded by the DUT and observed on G2.
- In Part G, the traffic transmitted from the broadcast IPv4 source address on G1 should not be forwarded by the DUT and not observed on G2.

Possible Problems: None.

Test AP.1.2: Source Address Denial

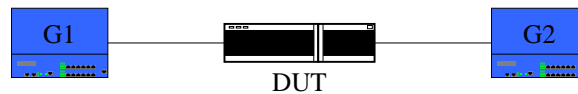
Purpose: To verify that a device properly denies source IPv6 addresses based on policy.

References: IPv6-SPEC

Resource Requirements: Monitor to capture packets, packet generators

Discussion: The primary building block for network access is accepting and denying application traffic based on source and destination IP addresses. The following test verifies that a router properly denies traffic sourced from a specific location. Do you have a specific quote from the Ipv6-SPEC that states that you must or must not accept the addresses?? This will make the testing easier to pass/fail.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny traffic originating from a global IPv6 Address

1. Configure the DUT to deny traffic from a globally defined source IPv6 address on its interface connected to G1.
2. From G1, transmit traffic to G2 containing the source IPv6 address configured in the previous step.
3. From G1, transmit traffic to G2 containing a global source address that is not denied per the configuration.
4. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic originating from a link local IPv6 Address

5. Configure the DUT to deny traffic from a link local defined source IPv6 address on its interface connected to G1.
6. From G1, transmit traffic to G2 containing the source IPv6 address configured in the previous step.
7. From G1, transmit traffic to G2 containing a global source address that is not denied per the configuration.
8. Observe the packets transmitted by the DUT on G2.

Part C: Deny traffic originating from a global IPv6 Prefix

9. Configure the DUT to deny traffic from a globally defined source IPv6 prefix on its interface connected to G1.
10. From G1, transmit traffic to G2 containing the source IPv6 prefix configured in the previous step.
11. From G1, transmit traffic to G2 containing a global source prefix that is not denied per the configuration.
12. Observe the packets transmitted by the DUT on G2.

Part D: Deny traffic originating from a link local IPv6 Prefix

13. Configure the DUT to deny traffic from a link local defined source IPv6 prefix on its interface connected to G1.
14. From G1, transmit traffic to G2 containing the source IPv6 prefix configured in the previous step.

15. From G1, transmit traffic to G2 containing a global source prefix that is not denied per the configuration.
16. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In all Parts, the traffic transmitted from denied addresses/networks on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the accepted address/networks at G1 should be forwarded by the DUT and observed on G2.

Possible Problems: The DUT may not allow the configuration of accept or deny function for the link-local address.

Test AP.1.3: Destination Address Denial

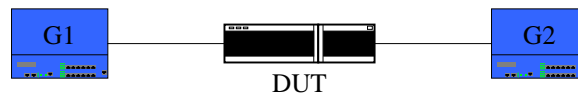
Purpose: To verify that a device properly denies destination IPv6 addresses based on policy.

References: IPv6-SPEC

Resource Requirements: Monitor to capture packets, generators

Discussion: The primary building block for network access is accepting and denying application traffic based on source and destination IP addresses. The following verifies that a router properly denies traffic destined to a specific location.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny traffic with a global IPv6 Address Destination

1. Configure the DUT to deny traffic from a globally defined destination IPv6 address on the interface connected to G1.
2. From G1, transmit traffic to G2 containing the destination IPv6 address configured in the previous step.
3. From G1, transmit traffic to G2 containing a destination address that is not denied per the configuration.
4. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic with a link local IPv6 Address Destination

5. Configure the DUT to deny traffic from a link local defined destination IPv6 address on the interface connected to G1.
6. From G1, transmit traffic to G2 containing the destination IPv6 address configured in the previous step.
7. From G1, transmit traffic to G2 containing a destination address that is not denied per the configuration.
8. Observe the packets transmitted by the DUT on G2.

Part C: Deny traffic with a global IPv6 Prefix

9. Configure the DUT to deny traffic from a globally defined destination IPv6 Prefix on the interface connected to G1.
10. From G1, transmit traffic to G2 containing the destination IPv6 prefix configured in the previous step.
11. From G1, transmit traffic to G2 containing a destination prefix that is not denied per the configuration.
12. Observe the packets transmitted by the DUT on G2.

Part D: Deny traffic all traffic with a local IPv6 prefix

13. Configure the DUT to deny traffic from a link local defined destination IPv6 prefix on the interface connected to G1.
14. From G1, transmit traffic to G2 containing the destination IPv6 prefix configured in the previous step.
15. From G1, transmit traffic to G2 containing a destination network prefix that is not denied per the configuration.
16. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In all Parts, the traffic transmitted from denied addresses/networks on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the accepted address/networks at G1 should be forwarded by the DUT and observed on G2.

Possible Problems: The DUT may not allow the configuration of accept or deny function for the link-local address.

Test AP.1.4: UDP Port Numbers

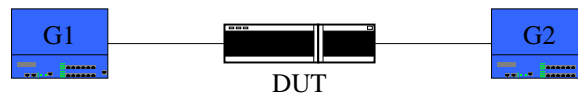
Purpose: To verify that a device properly accepts and denies UDP port numbers based on policy.

References: UDP

Resource Requirements: Monitor to capture packets, generators

Discussion: An extended building block for network access is accepting and denying application traffic based on source and destination UDP ports. The following verifies that a router properly accepts traffic to and from a UDP port.

Test Setup: Connect Devices as shown below



Procedure:

Part A: Deny traffic with a specific UDP Port Destination

1. Configure the DUT to deny traffic containing a destination UDP port on the interface connected to G1.
2. From G1, transmit traffic containing the destination UDP port configured in the previous step.
3. From G1, transmit traffic to G2 containing a valid destination UDP port.
4. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic with a specific UDP Port Source

5. Configure the DUT to deny traffic containing a source UDP port on the interface connected to G1.
6. From G1, transmit traffic to G2 containing the source UDP port configured in the previous step.
7. From G1, transmit traffic to G2 containing a valid source UDP port.
8. Observe the packets transmitted by the DUT on G2.

Part C: Accept all traffic with a specific UDP Port Destination

9. Configure the DUT to accept all traffic containing a destination UDP port on the interface connected to G1. Would you want a destination for G2?
10. From G1, transmit traffic to G2 containing the destination UDP port configured in the previous step.
11. Observe the packets transmitted by the DUT on G2.

Part D: Accept all traffic with a specific UDP Port Source

12. Configure the DUT to accept all traffic containing a source UDP port on the interface connected to G1.
13. From G1, transmit traffic to G2 containing the source UDP port configured in the previous step.
14. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Parts A and B, the traffic transmitted from denied UDP port on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the valid UDP port at G1 should be forwarded by the DUT and observed on G2.

- In Parts C and D, the traffic transmitted from G1 should be forwarded by the DUT and observed on G2.

Possible Problems: None.

Test AP.1.5: TCP Port Numbers

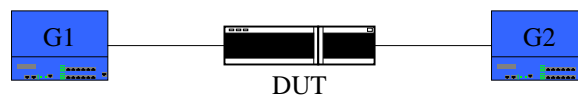
Purpose: To verify that a device properly accepts and denies TCP port numbers based on policy.

References: TCP

Resource Requirements: Monitor to capture packets, generators

Discussion: An extended building block for network access is accepting and denying application traffic based on source and destination TCP ports. The following verifies that a router properly accepts traffic to and from a TCP port.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny traffic with a specific TCP Port Destination

1. Configure the DUT to deny traffic containing a destination TCP port on the interface connected to G1.
2. From G1, transmit traffic to G2 containing the destination TCP port configured in the previous step.
3. From G1, transmit traffic to G2 containing a valid destination TCP port.
4. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic with a specific TCP Port Source

5. Configure the DUT to deny traffic containing a source TCP port on the interface connected to G1.
6. From G1, transmit traffic to G2 containing the source TCP port configured in the previous step.
7. From G1, transmit traffic to G2 containing a valid source TCP port.
8. Observe the packets transmitted by the DUT on G2.

Part C: Accept all traffic with a specific TCP Port Destination

9. Configure the DUT to accept all traffic containing a destination TCP port on the interface connected to G1.
10. From G1, transmit traffic to G2 containing the destination TCP port configured in the previous step.
11. Observe the packets transmitted by the DUT on G2.

Part D: Accept all traffic with a specific TCP Port Source

12. Configure the DUT to accept all traffic containing a source TCP port on the interface connected to G1.
13. From G1, transmit traffic to G2 containing the source TCP port configured in the previous step.
14. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Parts A and B, the traffic transmitted from denied TCP ports on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the valid TCP ports at G1 should be forwarded by the DUT and observed on G2.

- In Parts C and D, the traffic transmitted from G1 should be forwarded by the DUT and observed on G2.

Possible Problems: None.

Test AP.1.6: ICMPv6 Traffic

Purpose: To verify that a device properly accepts and denies ICMPv6 traffic based on policy.

References: ICMPv6

Resource Requirements: Monitor to capture packets, generators

Discussion: IPv6 nodes report errors encountered in processing packets and perform other internet-layer functions through the use of Internet Control Message Protocol for IPv6 (ICMPv6). These functions specifically include: (1) Destination Unreachable, (2) Packet Too Big, (3) Time Exceeded, (4) Parameter Problem, (128) Echo Request, (129) Echo Reply. The ICMPv6 functionality has been extended to have local meanings (133) Router Solicitation, (134) Router Advertisement, (135) Neighbor Solicitation, (136) Neighbor Advertisement and (137) Redirect. In the case of denying all ICMPv6 messages, if the DUT is delegating prefixes to local machines, the link-local ICMPv6 messages should be handled and exchanged with the DUT, even if the forwarding of ICMPv6 messages across the DUT is denied.

Test Setup: Connect Devices as shown below. If a stateful firewall is being tested, it must be configured for ICMPv6 stateful operation for Part E. This would deny all ICMP Echo Reply messages that are not paired with the same identifier as in the matching Echo Request. By default in this test with the exception of Part B, the firewall is configured to accept all types of ICMPv6 traffic.



Procedure:

Part A: Accept all ICMPv6 traffic

1. Configure the DUT to accept all ICMPv6 traffic on the interface connected to G1.
2. From G1, transmit traffic to G2 containing all possible ICMPv6 message types indicated in the type field of the ICMPv6 header.
3. Observe the packets transmitted by the DUT on G2.

Part B: Deny all ICMPv6 traffic

4. Configure the DUT to deny all ICMPv6 traffic on the interface connected to G1.
5. From G1, transmit traffic to G2 containing all other possible ICMPv6 message types.
6. Observe the packets transmitted by the DUT on G2.

Part C: Deny all ICMPv6 traffic, Link Local Prefix Delegation

7. Configure the DUT to deny all ICMPv6 traffic on the interface connected to G1.
8. Clear the Neighbor Discovery Cache on G1 and G2 either through configuration or a reboot.
9. From G1, transmit traffic to G2 containing all other possible ICMPv6 message types.
10. From G1, transmit UDP traffic to G2.
11. Observe the packets transmitted by the DUT on G2.

Part D: Deny ICMPv6 Echo Request messages, accept all other ICMPv6 Traffic

12. Configure the DUT to deny ICMPv6 Echo Request messages on the interface connected to G1.
13. From G1, transmit traffic to G2 containing all other possible ICMPv6 message types.
14. Observe the packets transmitted by the DUT on G2.

Part E: Accept ICMPv6 Echo Reply messages with an ID of Zero

15. Configure the DUT to deny ICMPv6 Echo Request messages with an ID number of zero on the interface connected to G1.
16. From G1, transmit traffic to G2 containing ICMPv6 Echo Replies with an ID of Zero.
17. From G1, transmit traffic to G2 containing ICMPv6 Echo Replies with a different ID number.
18. From G1, transmit traffic to G2 containing all other possible ICMPv6 message types.
19. Observe the packets transmitted by the DUT on G2.

Part F: Accept ICMPv6 Echo Reply messages with a non-Zero ID

20. Configure the DUT to deny ICMPv6 Echo Request messages with non-zero ID numbers on the interface connected to G1.
21. From G1, transmit traffic to G2 containing ICMPv6 Echo Replies with the ID numbers transmitted in Step 19.
22. From G1, transmit traffic to G2 containing ICMPv6 Echo Replies with ID numbers that differ from those transmitted in Step 19.
23. From G1, transmit traffic to G2 containing all other possible ICMPv6 message types.
24. Observe the packets transmitted by the DUT on G2.

Part G: Deny ICMPv6 Echo Reply messages, Accept all other ICMPv6 Traffic

25. Configure the DUT to deny all ICMPv6 Echo Reply messages.
26. From G1, transmit traffic to G2 containing multiple ICMPv6 Echo Requests and Replies with matching ID numbers.
27. From G1, transmit traffic to G2 containing all other possible ICMPv6 message types.
28. Observe the packets transmitted by the DUT on G2.

Part H: Accept ICMPv6 PMTU messages

29. Configure the DUT to accept ICMPv6 Path MTU messages on the interface connected to G1.
30. From G2, transmit ICMPv6 "Packet Too Big" message to G1.
31. From G1, transmit ICMPv6 "Packet Too Big" message to G2.
32. Observe the packets transmitted on G1 and G2.

Observable Results:

- In Part A, the ICMPv6 traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part B, the ICMPv6 traffic transmitted from G1 should be not forwarded by the DUT and not observed on G2.
- In Part C, the ICMPv6 traffic transmitted from G1 should be not forwarded by the DUT and not observed on G2. The router discovery process properly initializes and each generator receives a proper IPv6 address from the DUT. The UDP traffic is properly forwarded from G1 to G2.
- In Part D, the ICMPv6 Echo Request messages transmitted from G1 should not be forwarded by the DUT and not observed on G2. All other ICMPv6 messages transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Parts E and F, Echo Replies should not be accepted without a previous matching request. The ICMPv6 Echo Reply messages transmitted from G1 should not be forwarded by the DUT and not observed on G2. All other ICMPv6 messages transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part G, The ICMPv6 Echo Reply messages transmitted from G1 should not be forwarded by the DUT and not observed on G2. All other ICMPv6 messages transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part H, both Path MTU messages are properly forwarded by the DUT.

Possible Problems: There may be special configuration options for mobility. These options are covered in the next test. The DUT may not do prefix delegation, but if this is a transparent operation, the link local ICMPv6 should be allowed through to configure local devices. A device may not implement a stateful firewall and thus parts E and F may not be able to be tested.

Test AP.1.7: ICMPv6 Control Traffic for Mobile IPv6

Purpose: To verify that a device properly accepts and denies ICMPv6 control traffic for Mobile IPv6 based on policy.

References: ICMPv6
RFC 3775

Resource Requirements: Monitor to capture packets, generators

Discussion: IPv6 nodes report errors encountered in processing packets and perform other internet-layer functions through the use of Internet Control Message Protocol for IPv6 (ICMPv6). These functions specifically include Mobile IPv6. Mobile IPv6 is proposing to use values 144-147 and 150-153.

ICMPv6 type value 144 is the ICMP Home Agent Address Discovery (HAAD) Request message

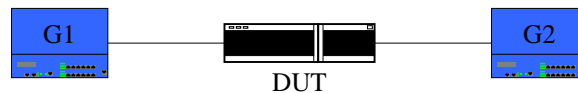
ICMPv6 type value 145 is the ICMP Home Agent Address Discovery (HAAD) Reply Message

ICMPv6 type value 146 is the ICMP Mobile Prefix Solicitation Message

ICMPv6 type value 147 is the ICMP Mobile Prefix Advertisement Message

While the values of 150-153 are not yet approved by the IANA, both sets of values are reserved due to earlier implementations of Mobile IPv6.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny HAAD Request message, Accept HAAD Reply message

1. Configure the DUT to deny all HAAD Request traffic on the interface connected to G1.
2. Configure the DUT to accept all HAAD Reply traffic on the interface connected to G1.
3. From G1, transmit traffic to G2 containing proper HAAD request and HAAD reply messages.
4. Observe the packets transmitted by the DUT on G2.

Part B: Accept HAAD Request message, Deny HAAD Reply message

5. Configure the DUT to deny all HAAD Request traffic on the interface connected to G1.
6. Configure the DUT to accept all HAAD Reply traffic on the interface connected to G1.
7. From G1, transmit traffic to G2 containing proper HAAD request and HAAD reply messages.
8. Observe the packets transmitted by the DUT on G2.

Part C: Deny Mobile Prefix Solicitation message, Accept Mobile Prefix Advertisement message

9. Configure the DUT to deny all Mobile Prefix Solicitation traffic on the interface connected to G1.
10. Configure the DUT to accept all Mobile Prefix Advertisement traffic on the interface connected to G1.
11. From G1, transmit traffic to G2 containing proper HAAD request and HAAD reply messages.
12. Observe the packets transmitted by the DUT on G2.

Part D: Accept Mobile Prefix Solicitation message, Deny Mobile Prefix Advertisement message

13. Configure the DUT to accept all Mobile Prefix Solicitation traffic on the interface connected to G1.

14. Configure the DUT to deny all Mobile Prefix Advertisement traffic on the interface connected to G1.
15. From G1, transmit traffic to G2 containing proper HAAD request and HAAD reply messages.
16. Observe the packets transmitted by the DUT on G2.

Part E: Deny Types 150, 151, 152 and 153

17. Configure the DUT to deny message types 150, 151, 152 and 153 on the interface connected to G1.
18. From G1, transmit traffic to G2 containing proper HAAD request and HAAD reply messages.
19. Observe the packets transmitted by the DUT on G2.

Part F: Accept Types 150, 151, 152 and 153

20. Configure the DUT to accept message types 150, 151, 152 and 153 on the interface connected to G1.
21. From G1, transmit traffic to G2 containing proper HAAD request and HAAD reply messages.
22. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Part A, the HAAD request traffic transmitted from G1 should not be forwarded by the DUT and not observed on G2. The HAAD reply traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part B, the HAAD request traffic transmitted from G1 should be forwarded by the DUT and observed on G2. The HAAD reply traffic transmitted from G1 should not be forwarded by the DUT and not observed on G2.
- In Part C, the ICMPv6 Echo Request messages transmitted from G1 should not be forwarded by the DUT and not observed on G2. All other ICMPv6 messages transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part D, Echo Replies should not be accepted without a previous matching request. The ICMPv6 Echo Reply messages transmitted from G1 should not be forwarded by the DUT and not observed on G2. All other ICMPv6 messages transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part E, all messages are properly dropped by the DUT.
- In Part F, all messages are properly forwarded by the DUT.

Possible Problems: None.

Test AP.1.8: Hop-by-Hop Header

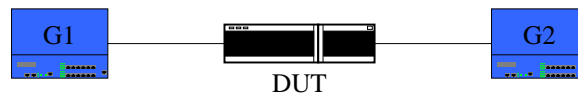
Purpose: To verify that a device properly denies traffic containing the Hop by Hop header.

References: ICMPv6
IPv6 Spec

Resource Requirements: Monitor to capture packets, generators

Discussion: There is a concern that the Hop-by-hop header may degrade the performance of existing networks if significant hop-by-hop traffic is encountered.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny all Hop-by-hop traffic

1. Configure the DUT to deny all hop-by-hop traffic on the interface connected to G1.
2. From G1, transmit a traffic mix to G2 containing UDP data with both hop-by-hop headers and no hop-by-hop headers.
3. Observe the packets transmitted by the DUT on G2.

Part B: Hop-by-hop screen

4. Configure the DUT to deny hop-by-hop traffic beyond a certain rate on the interface connected to G1.
5. From G1, transmit a traffic mix to G2 containing UDP data with both hop-by-hop headers and no hop-by-hop headers. Ensure that the hop-by-hop traffic is below the configured rate above.
6. From G1, transmit a traffic mix below the configured rate above to G2 containing UDP data with both hop-by-hop headers and no hop-by-hop headers. Ensure that the hop-by-hop traffic is above the configured rate above.
7. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Part A, the hop-by hop traffic transmitted from G1 should not be forwarded by the DUT and not observed on G2.
- In Part B, the hop-by hop traffic transmitted from G1 should be forwarded by the DUT and observed on G2. At Step 6, the hop-by-hop traffic exceeds the allotted limit and this traffic should not be forwarded by the DUT and not observed on G2.

Possible Problems: The DUT may not support a hop-by-hop screen configuration option.

Test AP.1.9: Default Router

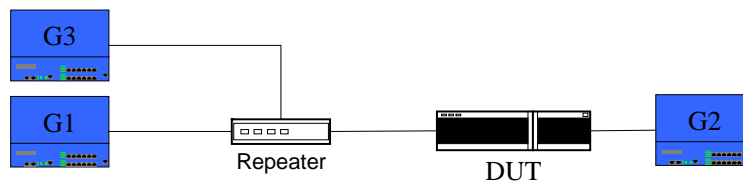
Purpose: To verify that a device properly denies traffic containing potentially harmful default router messages.

References: ICMPv6
IPv6 Spec
ND

Resource Requirements: Monitor to capture packets, generators

Discussion: There is a concern that a rogue IPv6 upstream router could declare itself the default router and hijack all upstream traffic. By default, a firewall should prevent this from occurring through denial of redirect messages from upstream routers and not allowing dynamic renumbering of the upstream network.

Test Setup: Connect Devices as shown below. Throughout the test, G1 continues to send valid router advertisements with a fixed prefix



Procedure:

Part A: Redirect Message

1. Allow the DUT to obtain a default route from G1.
2. From G2, transmit a traffic mix to G1 containing UDP data.
3. Connect G3 to the link between G1 and the DUT. G3 sends a valid redirect message with the source address of G1 to the DUT, signaling upstream traffic to be sent to it.
4. From G2, transmit a traffic mix to G1 containing UDP data.
5. Observe the packets transmitted by the DUT on G2.

Part B: Network Renumbering

6. Allow the DUT to obtain a default route from G1.
7. From G2, transmit a traffic mix to G1 containing UDP data.
8. Connect G3 to the link between G1 and the DUT. G3 sends a valid router advertisement with the source address of G1 and the valid and preferred prefix lifetimes of zero to the DUT. G3 sends a valid router advertisement as G3 with valid and preferred prefix lifetimes of a much larger number than zero.
9. From G2, transmit a traffic mix to G1 containing UDP data.
10. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Parts A and B, the DUT should ignore the messages from G3 and continue to forward the data to G1.

Possible Problems: The DUT may only support passive mode and thus the stations behind the DUT will be obtaining the default route and the IPv6 addresses. A passive device should also block malicious messages in this scenario.

Test AP.1.10: Time Based Authorization

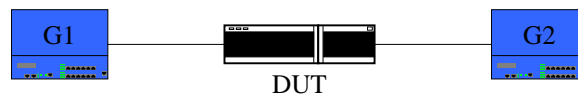
Purpose: To verify that a device properly denies source and IPv6 addresses based on time.

References: IPv6-SPEC

Resource Requirements: Monitor to capture packets, generators

Discussion: The primary building block for network access is accepting and denying application traffic based on source and destination IP addresses. Functionality can be increased if this can be extended to a specific access time.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny all traffic with a global IPv6 Source Address for a specific time

1. Configure the DUT to deny all traffic from a globally defined source IPv6 address on the interface connected to G1 for 2 minutes from the current time.
2. From G1, transmit traffic to G2 containing the source IPv6 address configured in the previous step.
3. From G1, after two minutes has expired, transmit traffic to G2 containing the source IPv6 address.
4. Observe the packets transmitted by the DUT on G2.

Part B: Deny all traffic with a global IPv6 Destination Address for a specific time

5. Configure the DUT to deny all traffic from a globally defined destination IPv6 address on the interface connected to G1 for 2 minutes from the current time.
6. From G1, transmit traffic to G2 containing the destination IPv6 address configured in the previous step.
7. From G1, after two minutes has expired, transmit traffic to G2 containing the destination IPv6 address.
8. Observe the packets transmitted by the DUT on G2.

Part C: Deny all traffic with a global IPv6 Source Network Address for a specific time

9. Configure the DUT to deny all traffic from a globally defined source IPv6 network on the interface connected to G1 for 2 minutes from the current time.
10. From G1, transmit traffic to G2 containing the source IPv6 address configured in the previous step.
11. From G1, after two minutes has expired, transmit traffic to G2 containing the source IPv6 network address.
12. Observe the packets transmitted by the DUT on G2.

Part D: Deny all traffic with a global IPv6 Destination Network Address for a specific time

13. Configure the DUT to deny all traffic from a globally defined destination IPv6 network on the interface connected to G1 for 2 minutes from the current time.
14. From G1, transmit traffic to G2 containing the destination IPv6 address configured in the previous step.
15. From G1, after two minutes has expired, transmit traffic to G2 containing the global IPv6 network address.
16. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In all Parts, the traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2 until the two-minute configuration time expires. After two minutes, traffic transmitted from G1 should be forwarded by the DUT and observed on G2.

Possible Problems: None.

Test AP.1.11: IPSec Forwarding

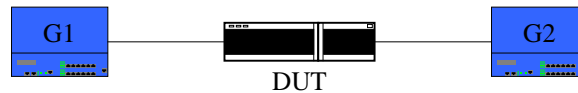
Purpose: To verify that a device will properly forward IPSec traffic.

References:

Resource Requirements: Monitor to capture packets, generators

Discussion: IPSec traffic should be setup to be forwarded end-to-end. A device should have the configuration flexibility to support a rule that allows or denies traffic with a next header of 50.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Accept traffic with a next header of 50 (IPSec)

1. Configure the DUT to accept traffic with a next header of 50 on the interface connected to G1. Configure the DUT to deny traffic with a next header of 6 (TCP) on the interface connected to G1.
2. From G1, transmit traffic to G2 containing a next header of 50.
3. From G1, transmit traffic to G2 containing a next header of 6.
4. From G1, transmit traffic to G2 containing a valid destination address range.
5. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic with a next header of 50 (IPSec)

6. Configure the DUT to deny traffic with a next header of 50 on the interface connected to G1. Configure the DUT to accept traffic with a next header of 6 (TCP) on the interface connected to G1.
7. From G1, transmit traffic to G2 containing a next header of 50.
8. From G1, transmit traffic to G2 containing a next header of 6.
9. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Part A, the traffic transmitted from G1 with a next header of 50 should be forwarded by the DUT and observed on G2. The traffic transmitted from G1 with a next header of 6 should not be forwarded by the DUT and should not be observed on G2.
- In Part B, the traffic transmitted from G1 with a next header of 50 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from G1 with a next header of 6 should be forwarded by the DUT and observed on G2.

Possible Problems: None.

Test AP.1.12: MAC Authorization

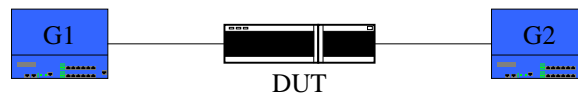
Purpose: To verify that a device has basic MAC access policy functionality.

References: IEEE802

Resource Requirements: Monitor to capture packets, generators

Discussion: Basic MAC address filtering functionality includes the ability to accept and deny source and destination address pairs from a layer 2 ethernet network perspective.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny traffic with a MAC Unicast Destination Address

1. Configure the DUT to deny traffic from a range of destination MAC unicast addresses on the interface connected to G1.
2. From G1, transmit traffic to G2 containing the destination MAC unicast address range configured in the previous step.
3. From G1, transmit traffic to G2 containing a valid destination address range.
4. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic with a MAC Unicast Source Address

5. Configure the DUT to deny traffic from a range of source MAC unicast addresses on the interface connected to G1.
6. From G1, transmit traffic to G2 containing the source MAC unicast address range configured in the previous step.
7. From G1, transmit traffic to G2 containing a valid source address range.
8. Observe the packets transmitted by the DUT on G2.

Part C: Deny traffic with a MAC Multicast Destination Address

9. Configure the DUT to deny traffic from a range of destination MAC multicast addresses on the interface connected to G1.
10. From G1, transmit traffic to G2 containing the destination MAC multicast address range configured in the previous step.
11. From G1, transmit traffic to G2 containing a valid destination multicast address range.
12. Observe the packets transmitted by the DUT on G2.

Part D: Deny traffic with a MAC Multicast Source Address

13. Configure the DUT to deny traffic from a range of source MAC multicast addresses on the interface connected to G1.
14. From G1, transmit traffic to G2 containing the source MAC multicast address range configured in the previous step.
15. From G1, transmit traffic to G2 containing a valid source address range.
16. Observe the packets transmitted by the DUT on G2.

Part E: Deny all traffic with a MAC Broadcast Destination Address

17. Configure the DUT to deny all traffic from the MAC broadcast address on the interface connected to G1.
18. From G1, transmit traffic to G2 containing the MAC broadcast address.
19. Observe the packets transmitted by the DUT on G2.

Part F: Accept all traffic with a MAC Broadcast Destination Address

20. Configure the DUT to accept all traffic from the MAC broadcast address on the interface connected to G1.
21. From G1, transmit traffic to G2 containing the MAC broadcast address.
22. Observe the packets transmitted by the DUT on G2.

Part G: Deny all traffic with a MAC Broadcast Source Address

23. Configure the DUT to deny all traffic from an MAC Broadcast Source Address on the interface connected to G1.
24. From G1, transmit traffic to G2 containing an MAC Broadcast Source Address.
25. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Parts A and C, the traffic transmitted from denied destination address range on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the other addresses at G1 should be forwarded by the DUT and observed on G2.
- In Parts B and D, the traffic transmitted from denied source address range on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the other addresses at G1 should be forwarded by the DUT and observed on G2.
- In Part E, the traffic transmitted from the broadcast IPv4 destination address on G1 should not be forwarded by the DUT and not be observed on G2.
- In Part F, the traffic transmitted from the broadcast IPv4 destination address on G1 should be forwarded by the DUT and observed on G2.
- In Part G, the traffic transmitted from the broadcast IPv4 source address on G1 should not be forwarded by the DUT and not observed on G2.

Possible Problems: None.

Test AP.1.13: ICMPv6 Destination Unreachable

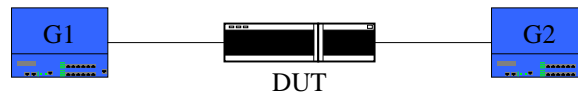
Purpose: To verify that a device properly sends an ICMPv6 destination unreachable message back to the client.

References: ICMPv6
ND

Resource Requirements: Monitor to capture packets, generators

Discussion: The HTTP server is connected over IPv4 and HTTP client is connected over both IPv4 and IPv6 (dual stack). When the HTTP client tries to establish an HTTP session over IPv6, it will receive an ICMPv6 error (destination unreachable) from the DUT. This ICMPv6 error will then become a trigger for the HTTP client to retry HTTP session over IPv4.

Test Setup: Connect Devices as shown below. Throughout the test, G1 continues to send valid router advertisements with a fixed prefix. G1 is configured to support both IPv4 and IPv6 (dual stack). G2 is configured only to support IPv4. Both links on the DUT are configured to support both IPv4 and IPv6 (dual stack). Configure the DUT to send an ICMPv6 Error in reply to a server query that is unreachable.



Procedure:

1. Setup IPv4 hostname for HTTP server on G2.
2. Setup G1 to establish an HTTP session for an IPv6 address matching the link on which G2 is connected.
3. Setup G1 to establish an HTTP session for the IPv4 address for G2.
4. Observe the packet exchanges on G1 and G2.

Observable Results:

- In Step 2, the DUT should send ICMPv6 error message of destination unreachable to G1. This will then trigger G1 to establish IPv4 HTTP session with G2.

Possible Problems: The DUT may not support configuration of the ICMPv6 Error message.

GROUP 2: Advanced AP Functionality

Scope:

The following tests are designed to verify the functionality and operation of mixed IPv4 and IPv6 based access authorization and other advanced features.

Overview:

Test AP.2.1: Combination Authorization

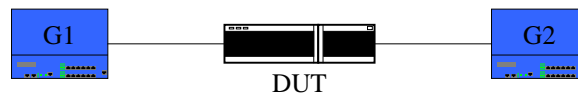
Purpose: To verify that a device properly accepts and denies traffic based on multiple rules.

References: IPv6-SPEC

Resource Requirements: Monitor to capture packets, generators

Discussion: Multiple rules for traffic acceptance and denial allow a device to accept and deny traffic for a complex authorization policy.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Time, Source Address, Destination Address

1. Configure the DUT to deny traffic from a globally defined source and destination IPv6 address pair on the interface connected to G1 for 2 minutes from the current time.
2. From G1, transmit traffic to G2 containing a variety of source and destination address pairs. Include packets that match the packet specifications that were configured in the previous step.
3. From G1, after two minutes, transmit traffic to G2 containing a variety of source and destination address pairs.
4. Observe the packets transmitted by the DUT on G2.

Part B: Time, UDP, Source Address, Destination Address

5. Configure the DUT to deny traffic from a globally defined source and destination IPv6 address pair on the interface connected to G1 for 2 minutes from the current time.
6. Configure the DUT to deny traffic containing a source UDP port on the interface connected to G1.
7. From G1, transmit traffic to G2 containing a variety of source and destination address pairs and UDP ports. Include packets that match the packet specifications that were configured in steps 5 and 6.
8. From G1, after two minutes, transmit traffic containing a variety of source and destination address pairs and UDP ports.
9. Observe the packets transmitted by the DUT on G2.

Part C: Time, TCP, Source Address, Destination Address

10. Configure the DUT to deny traffic from a globally defined source and destination IPv6 address pair on the interface connected to G1 for 2 minutes from the current time.
11. Configure the DUT to deny traffic containing a source TCP port on the interface connected to G1.

From G1, transmit traffic to G2 containing a variety of source and destination address pairs and TCP ports. Include packets that match the packet specifications that were configured in steps 10 and 11

12. From G1, after two minutes, transmit traffic to G2 containing a variety of source and destination address pairs and TCP ports.

13. Observe the packets transmitted by the DUT on G2.

Part D: Time, ICMPv6, Source Address, Destination Address

14. Configure the DUT to deny traffic from a globally defined source and destination IPv6 address pair on the interface connected to G1 for 2 minutes from the current time.
15. Configure the DUT to deny ICMPv6 Echo Request messages on the interface connected to G1.
16. From G1, transmit traffic to G2 containing a variety of source and destination address pairs and a mix of ICMPv6 messages and non-ICMPv6 IPv6 traffic. Include packets that match the packet specifications that were configured in steps 14 and 16.
17. From G1, after two minutes, transmit traffic to G2 containing a variety of source and destination address pairs and a mix of ICMPv6 messages and non-ICMPv6 IPv6 traffic.
18. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Part A, the [source, destination address pair] traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2 until the two-minute configuration time expires. After two minutes, the traffic transmitted from G1 should be forwarded by the DUT and observed on G2. All background traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part B, the [source, destination address pair and UDP port] traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2 until the two-minute configuration time expires. After two minutes, the traffic transmitted from G1 should be forwarded by the DUT and observed on G2. All background traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part C, the [source, destination address pair and TCP port] traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2 until the two-minute configuration time expires. After two minutes, the traffic transmitted from G1 should be forwarded by the DUT and observed on G2. All background traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part D, the [source, destination address pair and ICMPv6 Echo Request message] traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2 until the two-minute configuration time expires. After two minutes, the traffic transmitted from G1 should be forwarded by the DUT and observed on G2. All background traffic transmitted from G1 should be forwarded by the DUT and observed on G2.

Possible Problems: None.

Test AP.2.2: Ordered List Policy

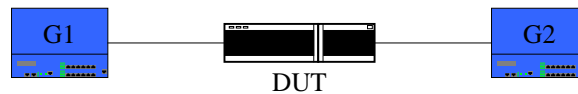
Purpose: To verify that a device properly implements an ordered list policy procedure.

References: IPv6-SPEC, TCP

Resource Requirements: Monitor to capture packets, generators
Monitor to capture packets, G2 can reply to ICMPv6 Echo Requests

Discussion: Access policies usually are in an ordered list and first match rule applies. To test this, a more specific deny rule is defined first and a more generic permit rule is defined second. It is ensured that the traffic matched the specific rules is dropped. The opposite scenario will ensure the traffic is permitted.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Specific Deny, Generic Allow

1. As the first item on the ordered configuration list, configure the DUT to deny traffic containing a specific destination TCP port on the interface connected to G1.
2. As the second item on the ordered configuration list, configure the DUT to accept all TCP traffic on the interface connected to G1.
3. From G1, transmit traffic to G2 containing the destination TCP port configured in the Step 1.
4. From G1, transmit traffic to G2 containing a valid destination TCP port.
5. Observe the packets transmitted by the DUT on G2.

Part B: *Generic Deny, Specific Allow*

6. As the first item on the ordered configuration list, configure the DUT to accept all TCP traffic on the interface connected to G1.
7. As the second item on the ordered configuration list, configure the DUT to deny traffic containing a specific destination TCP port on the interface connected to G1.
8. From G1, transmit traffic to G2 containing the destination TCP port configured in the Step 7.
9. From G1, transmit traffic to G2 containing a valid destination TCP port.
10. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Part A, the traffic transmitted from denied TCP port on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from other TCP ports at G1 should be forwarded by the DUT and observed on G2.
- In Part B, all traffic transmitted at G1 should be forwarded by the DUT and observed on G2, regardless of the TCP port.

Possible Problems: None.

Test AP.2.3: IPv4 and IPv6 Functionality, Same Policy

Purpose: To verify that a device can block a service with IPv4 and IPv6.

References: IP

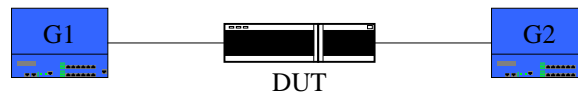
TCP

UDP

Resource Requirements: Monitor to capture packets, generatorsMonitor to capture packets

Discussion: Basic IPv4 functionality includes the ability to accept and deny source and destination address pairs on an IPv4 network.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny traffic with a specific UDP Destination Port

1. Configure the DUT to deny traffic containing a destination UDP port on the interface connected to G1.
2. From G1, transmit IPv6 and IPv4 traffic to G2 containing the destination UDP port configured in the previous step.
3. From G1, transmit IPv6 and IPv4 traffic to G2 containing a valid destination UDP port.
4. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic with a specific UDP Source Port

5. Configure the DUT to deny traffic containing a source UDP port range on the interface connected to G1.
6. From G1, transmit IPv6 and IPv4 traffic to G2 containing the destination UDP port configured in the previous step.
7. From G1, transmit IPv6 and IPv4 traffic to G2 containing a valid destination UDP port.
8. Observe the packets transmitted by the DUT on G2.

Part C: Deny traffic with a specific TCP Destination Port

9. Configure the DUT to deny traffic containing a destination TCP port range on the interface connected to G1.
10. From G1, transmit IPv6 and IPv4 traffic to G2 containing the destination TCP port configured in the previous step.
11. From G1, transmit IPv6 and IPv4 traffic to G2 containing a valid destination TCP port.
12. Observe the packets transmitted by the DUT on G2.

Part D: Deny traffic with a specific Source TCP Port

13. Configure the DUT to deny traffic containing a source TCP port range on the interface connected to G1.
14. From G1, transmit IPv6 and IPv4 traffic to G2 containing the destination TCP port configured in the previous step.
15. From G1, transmit IPv6 and IPv4 traffic to G2 containing a valid destination TCP port.
16. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Parts A and B, the traffic transmitted from denied UDP port on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the valid UDP ports at G1 should be forwarded by the DUT and observed on G2.
- In Parts C and D, the traffic transmitted from denied TCP port on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the valid TCP ports at G1 should be forwarded by the DUT and observed on G2.

Possible Problems: None.

Test AP.2.4: IPv4 and IPv6 Functionality, Different Policy

Purpose: To verify that a device can block a different services with IPv4 and IPv6.

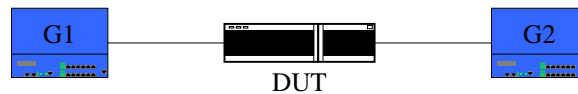
References: IP

TCP
UDP
IPv6-SPEC

Resource Requirements: Monitor to capture packets, generators

Discussion: Basic IPv4 functionality includes the ability to accept and deny source and destination address pairs on an IPv4 network.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Deny traffic with a specific UDP and TCP Destination Port

1. Configure the DUT to deny traffic containing a destination UDP port for IPv4 on the interface connected to G1.
2. Configure the DUT to deny traffic containing a destination TCP port for IPv6 on the interface connected to G1
3. From G1, transmit IPv4 traffic to G2 containing the destination UDP port configured in Step 1.
4. From G1, transmit IPv4 traffic to G2 containing the destination TCP port configured in Step 2.
5. From G1, transmit IPv6 and IPv4 traffic to G2 containing a valid destination UDP and TCP ports and addresses than described in Steps 4 and 5.
6. Observe the packets transmitted by the DUT on G2.

Part B: Deny traffic with a specific UDP and TCP Source Port

7. Configure the DUT to deny traffic containing a source UDP port for IPv4 on the interface connected to G1.
8. Configure the DUT to deny traffic containing a source TCP port for IPv6 on the interface connected to G1
9. From G1, transmit IPv4 traffic to G2 containing the source UDP port configured in Step 7.
10. From G1, transmit IPv4 traffic to G2 containing the source TCP port configured in Step 8.
11. From G1, transmit IPv6 and IPv4 traffic to G2 containing a valid destination UDP and TCP port ranges and addresses than described in Steps 10 and 11.
12. Observe the packets transmitted by the DUT on G2.

Part C: Accept all traffic with a specific UDP, TCP and ICMP Echo Requests Source Port

13. Configure the DUT to accept all traffic containing source UDP port X for IPv4 on the interface connected to G1. Configure the DUT to deny all other IPv4 UDP traffic on the interface connected to G1.
14. Configure the DUT to accept all traffic containing source TCP port X for IPv4 on the interface connected to G1. Configure the DUT to deny all other IPv4 TCP traffic on the interface connected to G1.

15. Configure the DUT to accept all ICMP Echo Request messages for IPv4 on the interface connected to G1. Configure the DUT to deny all other ICMP messages for IPv4 on the interface connected to G1.
16. Configure the DUT to deny all IPv6 traffic on the interface connected to G1.
17. From G1, transmit IPv4 and IPv6 traffic to G2 containing the following:
 - a. The source TCP port configured in Step 13.
 - b. The source TCP port configured in Step 14.
 - c. ICMP message traffic.
18. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Parts A and B, the traffic transmitted from denied TCP or UDP port on G1 should not be forwarded by the DUT nor observed on G2. The traffic transmitted from the valid TCP or UDP ports at G1 should be forwarded by the DUT and observed on G2.
- In Part C, the IPv4 traffic transmitted from the accept policy (TCP or UDP ports, or ICMP messages other than Echo Request messages) from G1 should be forwarded by the DUT and observed on G2. The IPv4 traffic transmitted from the other TCP or UDP ports or ICMP Echo Request messages should not be forwarded by the DUT nor observed on G2. IPv6 traffic should not be forwarded by the DUT.

Possible Problems: None.

Test AP.2.5: Tiny Fragment Attack for IPv4 and IPv6

Purpose: To verify that a device can block potentially malicious fragments.

References: IP

TCP
UDP
IPv6-SPEC
RFC1858

Resource Requirements: Monitor to capture packets, generators
Monitor to capture packets

Discussion: Most access policy implementations do not inspect packets beyond the first fragment, as it is assumed that the first fragment contains both the IP header and the TCP header. Malicious packets can be divided into fragments that can be forwarded past an edge router into a network even if basic access policy functionality is configured on that router. As fragmentation occurs at the router in IPv4 and at the host in IPv6, this creates several different attack scenarios. The Tiny Fragment Attack scenario is where a device transmits the suspicious fields of the TCP header concealed in the second fragment. According to [TCP], “Every internet module must be able to forward a datagram of 68 octets without further fragmentation”. The way to prevent this attack is for the access policy to be configured to search each fragment one (the second fragment) and to drop the packet if the protocol is TCP and if the length is less than 68 octets. The alternative is that if this condition is observed to drop fragment one (the second fragment).

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Valid Transmission of small IPv4 and IPv6 packets including fragments

1. Configure the DUT to deny all IPv4 and IPv6 traffic containing a fragment of zero in which the protocol is TCP and the length is less than 68 octets on the interface connected to G1.
2. From G1, transmit IPv4 and IPv6 traffic to G2 containing a set of properly formatted IPv4 and IPv6 packet fragment sequences in which the first fragment is 68 octets.
3. From G1, transmit IPv4 and IPv6 traffic to G2 containing a set of 64-byte IPv4 and IPv6 packets.
4. Observe the packets transmitted by the DUT on G2.

Part B: Tiny Fragment Attack for IPv4

5. Configure the DUT to deny all IPv4 traffic containing a fragment of zero in which the protocol is TCP and the length is less than 68 octets on the interface connected to G1.
6. From G1, transmit IPv4 and IPv6 traffic to G2 containing:
 - A first fragment that that contains 32 bits. This includes the IP header, the TCP source and destination port, a proper TCP sequence number and a proper TCP acknowledgement number.
 - A second fragment that has the TCP flags field with the SYN bit set.

7. Observe the packets transmitted by the DUT on G2.

Part C: Tiny Fragment Attack for IPv6

8. Configure the DUT to deny all IPv6 traffic containing a fragment of zero in which the protocol is TCP and the length is less than 68 octets on the interface connected to G1.

9. From G1, transmit IPv4 and IPv6 traffic to G2 containing:

- A first fragment that contains 32 bits. This includes the IP header, the TCP source and destination port, a proper TCP sequence number and a proper TCP acknowledgement number.
- A second fragment that has the TCP flags field with the SYN bit set.

10. Observe the packets transmitted by the DUT on G2.

Part D: Tiny Fragment Attack for IPv4 and IPv6

11. Configure the DUT to deny all IPv4 and IPv6 traffic containing a fragment of zero in which the protocol is TCP and the length is less than 68 octets on the interface connected to G1.

12. From G1, transmit IPv4 and IPv6 traffic to G2 containing:

- A first fragment that contains 32 bits. This includes the IP header, the TCP source and destination port, a proper TCP sequence number and a proper TCP acknowledgement number.
- A second fragment that has the TCP flags field with the SYN bit set.

13. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Part A, the traffic transmitted from G1 should all be forwarded by the DUT and observed on G2.
- In Part B, the IPv4 traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2. The IPv6 traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part C, the IPv6 traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2. The IPv4 traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part D, both the IPv4 and the IPv6 traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2.

Possible Problems: Some devices may not have separate configurations for blocking IPv4 and IPv6 fragments.

Test AP.2.6: Overlapping Fragment Attack for IPv4 and IPv6

Purpose: To verify that a device can block potentially malicious fragments.

References: IP

TCP
UDP
IPv6-SPEC
RFC1858

Resource Requirements: Monitor to capture packets, generators

Discussion: Most access policy implementations do not inspect packets beyond the first fragment, as it is assumed that the first fragment contains both the IP header and the TCP header. Malicious packets can be divided into fragments that can be forwarded past an edge router into a network even if basic access policy functionality is configured on that router. As fragmentation occurs at the router in IPv4 and at the host in IPv6, this creates several different attack scenarios. The Overlapping Fragment Attack creates packet streams, in which the second fragment contains an offset that could potentially overwrite part of the first fragment, regardless of the size of the first fragment. The way to prevent this attack is for the access policy to be configured to search each “fragment one” (the second fragment) and to drop the packet if the protocol is TCP and if the offset is less than 16 octets. This will ensure that the TCP flags field is in the first fragment.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Valid Transmission of Overlapping Fragments

1. Configure the DUT to deny all IPv4 traffic containing a fragment of one in which the protocol is TCP and the offset is less than 16 octets on the interface connected to G1.
2. From G1, transmit IPv4 and IPv6 traffic to G2 containing a set of properly formatted IPv4 and IPv6 packet fragment sequences in which the second fragment has an offset of 16 octets.
3. Observe the packets transmitted by the DUT on G2.

Part B: Overlapping Fragment Attack for IPv4

4. Configure the DUT to deny all IPv4 traffic containing a fragment of one in which the protocol is TCP and the offset is less than 16 octets on the interface connected to G1.
5. From G1, transmit IPv4 and IPv6 traffic to G2 containing:
 - A first fragment that that contains a proper IP and TCP header.
 - A second fragment that contains an offset of 15 octets and a TCP flags field with the SYN bit set.
6. Observe the packets transmitted by the DUT on G2.

Part C: Overlapping Fragment Attack for IPv6

7. Configure the DUT to deny all IPv6 traffic containing a fragment of zero in which the protocol is TCP and the offset is less than 16 octets on the interface connected to G1.
8. From G1, transmit IPv4 and IPv6 traffic to G2 containing:
 - A first fragment that that contains a proper IP and TCP header.

- A second fragment that contains an offset of 15 octets and a TCP flags field with the SYN bit set.
9. Observe the packets transmitted by the DUT on G2.
- Part D: Overlapping Fragment Attack for IPv4 and IPv6*
10. Configure the DUT to deny all IPv4 and IPv6 traffic containing a fragment of zero in which the protocol is TCP and the offset is less than 16 octets on the interface connected to G1.
11. From G1, transmit IPv4 and IPv6 traffic to G2 containing:
- A first fragment that contains a proper IP and TCP header.
 - A second fragment that contains an offset of 15 octets and a TCP flags field with the SYN bit set.
12. Observe the packets transmitted by the DUT on G2.

Observable Results:

- In Part A, the IPv4 and IPv6 traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part B, the IPv4 traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2. The IPv6 traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part C, the IPv6 traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2. The IPv4 traffic transmitted from G1 should be forwarded by the DUT and observed on G2.
- In Part D, both the IPv4 and the IPv6 traffic transmitted from G1 should not be forwarded by the DUT nor observed on G2.

Possible Problems: Some devices may not have separate configurations for blocking IPv4 and IPv6 fragments.

Test AP.2.7: Route Distribution

Purpose: To verify that a device can avoid distribution of specific routes.

References: IP

TCP
UDP
IPv6-SPEC
RFC1858

Resource Requirements: Monitor to capture packets, generators

Discussion: Most access policy implementations can avoid distribution of certain subnets to external entities. This can be a good security measure to deny knowledge of parts of a network behind the DUT, while advertising others. This test can be run with OSPF, RIP and BGP, however the DUT may not support one or more of these protocols. This test is optional, however a device may fail this test if the configuration conflicts with what is observed on the network. For example the configuration indicates that specific subnets are not to be advertised and their advertisement is observed on the network and in the routing protocol database exchange.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Distribute Routes for IPv4

1. Configure an access policy on the DUT for the IPv4 address on the interface connected to G1.
2. Configure (who is them?? Be specific) not to distribute. Enable an IGP routing protocol to correspond with the configuration.
3. Disable the function configured in step 2.
4. You need to transmit packets before and after you disable the functions. Same for all parts in this test.
5. Observe the routing protocol control packets transmitted by the DUT on G2.

Part B: Distribute Routes for IPv6

6. Configure an access policy on the DUT for the global IPv6 address on the interface connected to G1.
7. Configure them not to distribute. Enable an IGP routing protocol to correspond with the configuration.
8. Disable the function configured in step 6.
9. Observe the routing protocol control packets transmitted by the DUT on G2.

Part C: Distribute Routes for IPv4 and IPv6

10. Configure an access policy on the DUT for the IPv4 address on the interface connected to G1.
11. Configure an access policy on the DUT for the IPv6 address on the interface connected to G1.
12. Configure them not to distribute. Enable an IGP routing protocol to correspond with the configuration.
13. Disable the function configured in step 11.
14. Observe the routing protocol control packets transmitted by the DUT on G2.

Observable Results:

- In all Parts, the routing protocol control traffic transmitted to G2 should not advertise the IP address information on the network connected directly to G1 and the DUT. Once this feature is disabled, advertisements should be observed.

Possible Problems: None.

Test AP.2.8: Long Access Policy List Forwarding

Purpose: To verify that a device does not degrade forwarding with a long Access Policy configuration.

References: IP
TCP
UDP
IPv6-SPEC
RFC1858

Resource Requirements: Monitor to capture packets, generators

Discussion: Most access policy implementations can handle complex rules in terms of their access policy definition. This may cause jitter in the packet forwarding if the filter is done in software.

Test Setup: Connect Devices as shown below.



Procedure:

Part A: Long access policy for IPv4

1. Configure a long access policy list for IPv4. Every other command on this list is a no-operation command and will not be referenced with the type of traffic that will be sent.
2. From G1, transmit a mixture of frames to G2 with a fixed inter-packet gap. The frames abide by different accept rules in the long list configured in step 1, and ensure the DUT does not reference the many no-operation commands.
3. Observe the packets received on G2.
4. Remove the no-operation commands from the list.
5. From G1, transmit a mixture of frames as in Step 2.
6. Observe the packets received on G2.

Part B: Long access policy for IPv6

7. Configure a long access policy list for IPv6. Every other command on this list is a no-operation command and will not be referenced with the type of traffic that will be sent.
8. From G1, transmit a mixture of frames to G2 with a fixed inter-packet gap. The frames abide by different accept rules in the long list configured in step 4, and ensure the DUT does not reference the many no-operation commands.
9. Observe the packets received on G2.
10. Remove the no-operation commands from the list.
11. From G1, transmit a mixture of frames as in Step 8.
12. Observe the packets received on G2.

Part C: Long access policy for IPv4 and IPv6

13. Configure a long access policy list for IPv4 and IPv6. Every other command on this list is a no-operation command and will not be referenced with the type of traffic that will be sent.
14. From G1, transmit a mixture of frames to G2 with a fixed inter-packet gap. The frames abide by different accept rules in the long list configured in step 1, and ensure the DUT does not reference the many no-operation commands.
15. Observe the packets received on G2.

16. Remove the no-operation commands from the list.
17. From G1, transmit a mixture of frames as in Step 14.
18. Observe the packets received on G2.

Observable Results:

- In all Parts, the forwarded frames should have a fixed inter-packet gap equivalent to the gap that was transmitted from G1. There should be no difference in the traffic forwarding between the list with the no-operation commands and the list without the no-operation commands.

Possible Problems: None.