



**ICSA Labs
Network Protection Devices
Test Specification
Version 1.1**

February 10, 2010

www.icsalabs.com

Change Log

Version 1.1 February 10, 2010

- added a step in test 1.6.1 instructing that settings be applied before the removal of power.
- fixed a typo in test 2.2.3 in which the word “request” was used instead of “response.”
- changed format of loopback address in 4.2.2a,d from “1” to the more common “::1”
- added step for unspecified address testing to test 2.1.1.3
- added specifics for multicast source address test in test 2.1.1.3
- added specifics for ipv4-mapped/compatible addresses to 2.1.1.3
- removed unique local requirement in test 2.1.1.3
- clarified the requirements and tests for 1.5.2
- clarified 4.2.2l detection of MTU below minimum
- replaced guidance for selecting relevant vulnerabilities with reference to USGv6 website
- removed configuration notes referring to fully loading the IDS/IPS from all except test 4.6
- altered test 4.1.3, 4.4.3 & 5.2.3 to include replaying traffic in addition to launching attack test cases manually

Version 1.0 November 19, 2009

- simplified example network infrastructure for firewall testing
- altered firewall tests to represent the changed infrastructure
- added section 2.5, tunneled traffic handling for firewall testing
- generalized IDS/IPS test configuration description to focus on capabilities hardware provides and not the actual type of hardware to be used to allow for flexibility in test bed design
- limited scope of IDS/IPS protection testing from all vulnerabilities in listed vendors' products to high-severity (i.e., CVSS score of at least 8)
- merged section 4.5 (Logging and Alerts) into section 4.1 (Known Attack Detection) and modified title of latter to reflect new scope
- corrected procedure item 2b in 4.2 from "source address" to "destination address" set to 0 (unspecified address)
- added table of contents
- corrected typos and minor format issues

Version 0.90 May 15, 2009

| | |
|------------------------------------------------------------------------------|-----------|
| REQUIREMENTS..... | 4 |
| 1. Common..... | 4 |
| 2. Firewalls..... | 5 |
| 3. Application Firewalls..... | 6 |
| 4. Intrusion Detection Systems..... | 7 |
| 5. Intrusion Prevention Systems..... | 8 |
| TEST PLAN..... | 9 |
| 1. Common Requirements | 9 |
| 1.1.1: Configuration of protective functionality..... | 9 |
| 1.2.1: Support Dual Stack..... | 10 |
| 1.3.1: Configuration of protective functionality..... | 11 |
| 1.3.2: Configuration of logging and alert facility configurations..... | 11 |
| 1.3.3: Selectively restricting rights to the administrative interface..... | 11 |
| 1.4.1: Selectively restricting rights to the administrative interface..... | 13 |
| 1.4.2: Individual rights..... | 13 |
| 1.5.1: Administrative access..... | 14 |
| 1.5.2: Administrative communications..... | 14 |
| 1.6.1: Persistence of device settings..... | 15 |
| 1.7.1, 1.7.2: Configuration change logging..... | 16 |
| 1.8.1: Proper handling of fragmented packets..... | 17 |
| 1.9.1: Handling of v4/v6 Tunneling Schemes..... | 17 |
| 2. Firewalls | 18 |
| 2.1.1.1: Allowing/Blocking IPv6 Packets Sent to the Firewall..... | 20 |
| 2.1.1.2: Allowing/Blocking IPv6 Packets Sent through the Firewall..... | 21 |
| 2.1.1.3: Illegal Source and Destination Addresses..... | 23 |
| 2.1.2.1: Selectively Block IPv6 packets based on Next Header values..... | 25 |
| 2.1.2.2: Type 0 Routing Headers..... | 26 |
| 2.1.3.1: TCP/UDP ports..... | 27 |
| 2.1.3.2: ICMPv6..... | 28 |
| 2.1.4.1: Implicit Deny Policy..... | 29 |
| 2.2.1: Asymmetrical Controls..... | 30 |
| 2.2.2: Allowing Connection Oriented Protocols..... | 31 |
| 2.2.3: Selectively Blocking External Replies from the external network:..... | 32 |
| 2.3.1: ESP and AH Traffic Handling:..... | 33 |
| 2.3.2: Establishing Security Associations with IPsec..... | 34 |
| 2.4.1: Fail-Safe..... | 35 |
| 2.5: Tunneled traffic handling:..... | 36 |
| 3. Application Firewalls | 37 |
| 3.1.1. Trust Barriers..... | 37 |
| 3.2.1. Session traffic authorization..... | 37 |
| 3.3.1. File filtering..... | 38 |
| 4. Intrusion Detection Systems | 39 |
| 4.1: Known attack detection & Logging and alerts..... | 39 |
| 4.2: Malformed packet detection..... | 40 |
| 4.3: Port-scanning detection..... | 42 |
| 4.4: Tunneled traffic detection..... | 43 |
| 4.5: Logging and alerts..... | 44 |
| 4.6: Performance under load, fail-safe..... | 45 |
| 5. Intrusion Prevention Systems | 46 |
| 5.1: Implement intrusion detection capabilities..... | 46 |
| 5.2: Stop or attenuate detected attacks & Logging and alerts..... | 46 |

Requirements

1. Common

- 1.1. Basic host or router IPv6 connectivity requirements
 - 1.1.1. NPDs may only implement basic IPv6 protocol capabilities necessary to perform their security function. No basic IPv6 connectivity requirements are specified here.
- 1.2. Dual stack
 - 1.2.1. The NPD MAY support both IPv4 and IPv6 protocols. Only IPv6 protection functionality is addressed.
- 1.3. Administrative functionality
 - 1.3.1. The NPD's administrative interface MUST have the ability to configure its protective functionality.
 - 1.3.2. The NPD's administrative interface MUST have the ability to modify its logging and alert facility configuration.
 - 1.3.3. The NPD MUST have the ability to selectively restrict rights to its administrative interface.
- 1.4. Authentication and authorization
 - 1.4.1. All administrative controls MUST be secure from non-authorized access and restricted to appropriately authorized users.
 - 1.4.2. In the case where separate user rights are offered, the NPD MUST enforce any individual rights applied to each user.
- 1.5. Security of control and communications
 - 1.5.1. All administrative controls MUST be secure from non-authorized access.
 - 1.5.2. All administrative communications with a NPD must be secure from outside observation. The NPD MUST support at least one of the following mechanisms and be capable of disabling any insecure administrative communications:
 - o local console access
 - o FIPS-approved encrypted network communication
 - o a separate channel that can be isolated from all other network traffic
- 1.6. Persistence
 - 1.6.1. All NPD settings MUST persist through loss and restoration of electrical power.
- 1.7. Logging and alerts
 - 1.7.1. NPDs MUST provide sufficient administrative capability to allow inspection of all administratively-controlled settings and give assurance of their proper functioning.
 - 1.7.2. This functionality MUST be controllable by, and accessible to, properly authorized administrators.
- 1.8. Fragmented packet handling
 - 1.8.1. NPDs MUST be able to handle fragmented packets, whether by provisionally reassembling and applying appropriate controls based on the reassembled packet, or (in the case of firewalls) by blocking fragments that cannot otherwise be handled.
- 1.9. Tunneled traffic handling
 - 1.9.1. The NPD MUST be able to handle all v4/v6 tunneling schemes, no matter how embedded, either by analyzing and applying the appropriate controls based on the encapsulated packet header, or (in the case of firewalls) by simply blocking all unanalyzed tunneled packets.

2. Firewalls

2.1. Port/protocol/address blocking

2.1.1. IPv6 addresses

2.1.1.1: the firewall **MUST** selectively allow/block any IPv6 packet sent to any of its interfaces based on either the source or destination address in the packet sent.

2.1.1.2: the firewall **MUST** selectively allow/block any IPv6 packet sent inbound or outbound through its interfaces based on either the source or destination address in the packet sent.

2.1.1.3: the firewall **MUST** be able to block packets which contain illegal source and destination addresses.

2.1.2. Extension header types

2.1.2.1: the firewall **MUST** selectively block any IPv6 packet based on its Next Header.

2.1.2.2: the firewall **MUST** have the ability to selectively block type 0 routing headers.

2.1.3. Upper layer protocols

2.1.3.1. TCP/UDP: the firewall **MUST** selectively allow/block TCP and UDP packets based on the set source or destination port.

2.1.3.2. ICMPv6: the firewall **MUST** selectively allow/block ICMPv6 traffic by type and code.

2.1.4. Implicit deny policy

2.1.4.1: the firewall **MUST** block any traffic that has not been explicitly allowed.

2.2. Asymmetrical blocking

2.2.1: the firewall **MUST** distinguish between internal and external networks and allow asymmetrical controls of traffic between these networks.

2.2.2: the firewall **MUST** allow connection oriented protocols such as TCP to travel bi-directionally.

2.2.3: the firewall **SHOULD** be able to selectively block unsolicited replies from the external network.

2.3. IPsec traffic handling

2.3.1: the firewall **MUST** selectively block ESP and AH traffic.

2.3.2: the firewall **MAY** be capable of establishing Security Associations with IPsec (security gateway).

2.4. Performance under load, fail-safe

2.4.1: When the firewall is suffering performance degradation due to overuse of resources it **MUST** fail in a manner that does not allow unauthorized access to itself or internal and external networks attached.

3. Application Firewalls

3.1. Violation of trust barriers

3.1.1. The application firewall **MUST NOT** violate trust barriers by either:

- rewriting incoming untrusted data to appear trusted, or
- exposing information (e.g. internal network structures) to external untrusted networks.

3.2. Session traffic authorization

3.2.1. The application firewall **MUST** have the capability for controlled authorization for establishing sessions from the external network to internal hosts.

3.3. Email, file filtering

3.3.1. The application firewall **MUST** have the capability to examine files (e.g. e-mail attachments) for malicious content and selectively block them. That is, the application firewall **MUST** provide sufficient means to block typical threat traffic.

4. Intrusion Detection Systems

4.1. Known attack detection

The NPD **MUST** be configurable to detect vulnerability-related attacks relevant to USG organizations.

4.2. Malformed packet detection

The NPD **MUST** detect many different kinds of malformed and non-standard IPv6 frames.

4.3. Port-scanning detection

The NPD **MUST** be capable of detecting TCP connect and UDP port scan traffic passing through the NPD.

4.4. Tunneled traffic detection

When IPv6 packets are tunneled through an NPD using a common form of encapsulation, the NPD **MUST** be able to detect vulnerability-related attacks relevant to USG organizations that have occurred in recent years. When attacks like this are passed through the NPD using more than a single layer of encapsulation, the NPD **MUST** either detect the attacks or detect that more than a single layer of encapsulation is present.

4.5. Logging and alerts

The NPD **MUST** be able to detect and log vulnerability-related attacks relevant to USG organizations that have occurred in recent years.

4.6. Performance under load, fail-safe

The NPD **MUST** demonstrate when configured to do so that it notifies administrators (be it via an alert, a log message, etc.) when under severe load.

5. Intrusion Prevention Systems

5.1. Implement detection (listed in previous section)

Perform ALL of the test cases in section 4.

5.2. Stop or attenuate detected attacks

The NPD (in this case a network IPS) **MUST** be able to block and log vulnerability-related attacks relevant to USG organizations that have occurred in recent years.

Test Plan

1. Common Requirements

1.1: Basic host or router IPv6 connectivity requirements

1.1.1: Configuration of protective functionality.

Requirement:

NPDs may only implement basic IPv6 protocol capabilities necessary to perform their security function. No basic IPv6 connectivity requirements are specified here.

Note:

Not tested as part of the NPD test plan. Refer to the applicable Conformance and Interoperability Test Selections.

1.2 Dual Stack

1.2.1: Support Dual Stack

Requirement:

The NPD MAY support both IPv4 and IPv6 protocols. Only IPv6 protection functionality is addressed.

Note:

Not tested as part of the NPD test plan. Refer to the applicable Conformance and Interoperability Test Selections.

1.3: Administrative Functionality

1.3.1: Configuration of protective functionality.

Requirement:

The NPD's administrative interface **MUST** have the ability to configure its protective functionality.

Test Configuration:

The default configuration according to the NPD type.

Procedure:

1. Add new rules to the NPD.
2. Remove rules from the NPD.
3. Alter rules on the NPD.

Results:

1. Verify that new rules added function as expected.
2. Verify that removed rules no longer function.
3. Verify that altered rules function as expected.

1.3.2: Configuration of logging and alert facility configurations.

Requirement:

The NPD's administrative interface **MUST** have the ability to modify its logging and alert facility configurations.

Test Configuration:

The default configuration according to the NPD type with logging facilities enabled.

Procedure:

1. Enable logging and alert facilities on all individual configurable items available.
2. Trigger log messages to be sent for each configurable item.
3. If able, change the logging destination.
4. Trigger log messages to be sent for each configurable item so that they may be sent to the new configured logging destination.

Results:

1. Verify that all logging features operate as configured.
2. If logging destination can be changed, verify that log messages are delivered the configuration destination.

1.3.3: Selectively restricting rights to the administrative interface.

Requirement:

The NPD **MUST** have the ability to selectively restrict rights to its administrative interface.

Test Configuration:

The default configuration according to the NPD type.

Procedure:

1. Attempt to connect to the various administrative interfaces (e.g., GUI, serial, ssh) necessary for testing, by using credentials of a user that does not exist or was not given administrative rights.

Results:

1. Verify that you cannot gain administrative access.

1.4 Authentication and authorization

1.4.1: Selectively restricting rights to the administrative interface.

Requirement:

All administrative controls **MUST** be secure from non-authorized access and restricted to appropriately authorized users.

Test Configuration:

The default configuration according to the NPD type.

Procedure:

1. Attempt to connect to the various administrative interfaces (e.g., GUI, serial, ssh) necessary for testing, by using credentials of a user that does not exist or was not given administrative rights.
2. Attempt to connect to the various administrative interfaces (e.g., GUI, serial, ssh) necessary for testing, by using credentials with no password.
3. Scan interfaces for any known vulnerabilities.
4. Attempt to gain access to the administrative interface through any found vulnerabilities.

Results:

1. Verify that you cannot gain administrative access via false authentication information.
2. Verify that no vulnerabilities allowed access into the administrative interfaces.

1.4.2: Individual rights

Requirement:

In the case where separate user rights are offered, the NPD **MUST** enforce any individual rights applied to each user.

Test Configuration:

The default configuration according to the NPD type.

Procedure:

1. Create two user accounts.
2. Apply different rights to each user.

Results:

1. Verify that the assigned individual rights operate as expected.

1.5: Security of control communications

1.5.1: Administrative access.

Requirement:

All administrative controls **MUST** be secure from non-authorized access.

Test Configuration:

Not separately tested. Refer to test 1.4.1.

1.5.2: Administrative communications.

Requirement:

The NPD **MUST** support at least one of the following mechanisms and be capable of disabling any insecure administrative communications:

- local console access
- FIPS-approved encrypted network communication
- a separate channel that can be isolated from all other network traffic

Test Configuration:

The default configuration according to the NPD type.

Procedure:

1. Configure the NPD to enable secure administrative communication using each of the secure mechanisms that the NPD supports, and disable any insecure administrative communication capabilities.
2. Perform administrative functions, such as configuring policies, reviewing logs, etc.

Results:

1. Verify that administration functions perform properly via the secure mechanisms.
2. Verify that any supported insecure mechanisms are disabled.
3. For encrypted network communications, verify that the NPD can be configured to use only FIPS-approved algorithms (see the most recent version of FIPS 140 Security Requirements for Cryptographic Modules).

1.6: Persistence

1.6.1: Persistence of device settings

Requirement:

All device settings **MUST** persist through loss and restoration of electrical power.

Test Configuration:

The default configuration according to the NPD type.

Procedure:

1. Configure firewall.
2. Apply the settings to the firewalls non volatile storage.
3. Remove power from the firewall.
4. Reapply power to the firewall.

Results:

1. Verify that the firewall did not lose authentication information.
2. Verify that the firewall kept all policy configurations.

1.7: Logging and alerts

1.7.1, 1.7.2: Configuration change logging

Requirement:

NPDs **MUST** provide sufficient administrative capability to authorized administrators allowing the inspection of all administratively-controlled settings and give assurance of their proper functioning.

Test Configuration:

The default configuration according to the NPD type.

Procedure:

1. Add a new policy rule.
2. Delete an existing policy rule.
3. Change an existing policy rule.
4. Change a user password.
5. Change the system time.

Results:

1. Verify that all procedures generate appropriate log messages and that they are viewable by only the authorized administrator.

1.8: Fragmented packet handling

1.8.1: Proper handling of fragmented packets

Requirement:

NPDs **MUST** be able to handle fragmented packets, whether by provisionally reassembling and applying appropriate controls based on the reassembled packet, or (in the case of firewalls) by blocking fragments that cannot otherwise be handled.

Test Configuration:

The default configuration according to the NPD type.

Procedure:

1. Attempt to send a fragmented packet through the NPD.

Results:

1. Verify that the NPD reassembles the packet or blocks it if reassembly is not possible.

1.9. Tunneled traffic handling

1.9.1. Handling of v4/v6 Tunneling Schemes

Requirement:

The NPD **MUST** be able to handle all v4/v6 tunneling schemes, no matter how embedded, either by analyzing and applying the appropriate controls based on the encapsulated packet header, or (in the case of firewalls) by simply blocking all unanalyzed tunneled packets.

Note:

Not tested separately. See Firewall (2.5) and Intrusion Detection/Protection System (4.4) testing sections.

2. Firewalls

Test network Infrastructure Configuration:

The test infrastructure used in section 2 of testing will be configured as seen in figure 1.

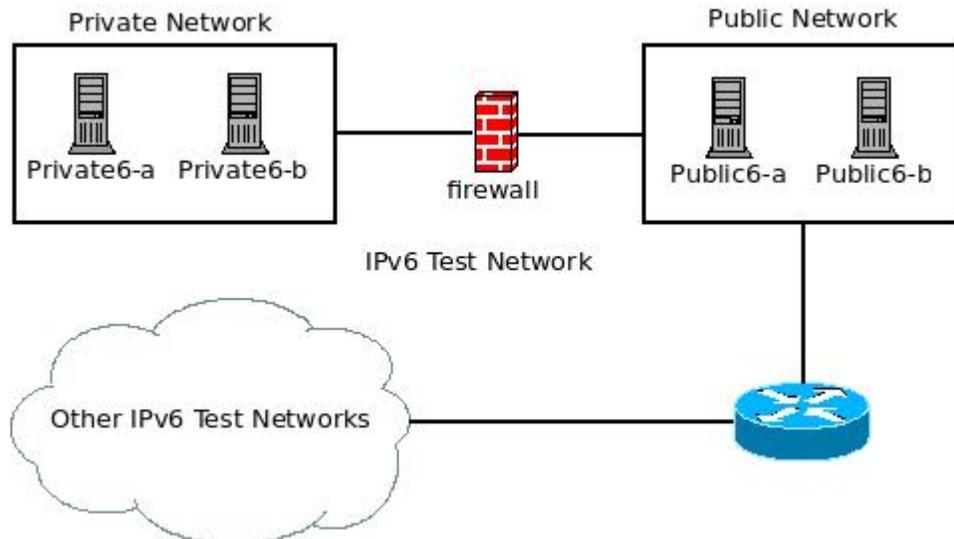


Figure 1 (Firewalls Testing Infrastructure Configuration)

Host Configuration:

All hosts in the test network must be configured with the following listening services in figure 2.

| Service | Protocol | Port |
|---------|----------|------|
| HTTP | TCP | 80 |
| HTTPS | TCP | 443 |
| SMTP | TCP | 25 |
| DNS | TCP | 53 |
| DNS | UDP | 53 |

Figure 2 (2.1.1 Testing Host Configuration)

Default Firewall Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public6 | Private6 | Any | 80 | TCP |
| Public6 | Private6 | Any | 53 | UDP |

Deny:

Default Deny

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|-----------------------|----------------------------|--------------------|-------------------------|-----------------|
| Host A1 | Host E1 | Any | 80 | TCP |
| Host A1 | Host E1 | Any | 53 | UDP |

Deny:

Default Deny

In addition to the above rules, the administrative interface must be configured and accessible.

2.1. Port/Protocol/Address Blocking

2.1.1.1: Allowing/Blocking IPv6 Packets Sent to the Firewall

Requirement:

The firewall MUST selectively allow/block any IPv6 packet sent to any of its interfaces based on either the source or destination address in the packet sent.

Test Configuration:

Allowed Configuration:

To external address from public network:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|-----------|
| Public6-a | Firewall | Any | Listening | Listening |

Deny:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|-----------|
| Public6-b | Firewall | Any | Listening | Listening |

To internal address from private network:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|-----------|
| Private6-a | Firewall | Any | Listening | Listening |

Deny:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|-----------|
| Private6-b | Firewall | Any | Listening | Listening |

Procedure:

1. Configure firewall as seen above.
2. Attempt to access an allowed port on the NPD from Public6-a.
3. Attempt to access a denied port on the NPD from Public6-b.
4. Attempt to access an allowed port on the NPD from Private6-a.
5. Attempt to access a denied port on the NPD from Private6-b.

Results:

1. Verify the allowed legitimate IPv6 traffic is properly accepted by the firewall.
2. Verify the denied IPv6 traffic is properly dropped by the firewall.

2.1.1.2: Allowing/Blocking IPv6 Packets Sent through the Firewall

Requirement:

The firewall MUST selectively allow/block any IPv6 packet sent inbound or outbound through its interfaces based on either the source or destination address in the packet sent.

Test Configuration:

1.) Allow out/in based on source address:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public6-a | Private net | Any | 80 | TCP |

Deny:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public6-b | Private net | Any | 80 | TCP |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private6-a | Public net | Any | 80 | TCP |

Deny:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private6-b | Public net | Any | 80 | TCP |

2.) Allow out/in based on destination address:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public net | Private6-a | Any | 80 | TCP |

Deny:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public net | Private6-b | Any | 80 | TCP |

Outbound:
Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private net | Public6-a | Any | 80 | TCP |

Deny:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private net | Public6-b | Any | 80 | TCP |

Procedure:

1. Configure the NPD as seen in “Allow out/in based on source address.”
2. Attempt to send allowed traffic inbound from Public6-a to a host on the private network.
3. Attempt to send denied traffic inbound from Public6-b to a host on the private network.
4. Attempt to send allowed traffic outbound from Private6-a to a host on the public network.
5. Attempt to send denied traffic outbound from Private6-b to a host on the public network.
6. Configure the NPD as seen in “Allow out/in based on destination address.”
7. Attempt to send allowed traffic inbound from a host on the public network to Private6-a.
8. Attempt to send denied traffic inbound from a host on the public network to Private6-b.
9. Attempt to send allowed traffic outbound from a host on the private network to Public6-a.
10. Attempt to send denied traffic outbound from a host on the private network to Public6-b.

Results:

1. Verify that the rules set on the NPD allow and block as they are written.

2.1.1.3: Illegal Source and Destination Addresses

Requirement:

The firewall MUST block any IPv6 packet sent inbound or outbound through its interfaces with an illegal source or destination address.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Any | Any | Any | Any | Any |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Any | Any | Any | Any | Any |

Procedure:

1. Test illegal packet with an Unspecified Address as the source
 - a. Send a packet with the source address set to an Unspecified Address (0:0:0:0:0:0:0:0) outbound and inbound through the product.
2. Test illegal packets with multicast addresses set as the source.
 - a. Send a packet with a source address of a multicast address from the interface-local scope outbound and inbound through the firewall.
 - b. Send a packet with a source address of a multicast address from the link-local scope outbound and inbound through the firewall.
 - c. Send a packet with a source address of a multicast address from the admin-local scope outbound and inbound through the firewall.
 - d. Send a packet with a source address of a multicast address from the site-local scope outbound and inbound through the firewall.
 - e. Send a packet with a source address of a multicast address from the organization-local scope outbound and inbound through the firewall.
3. Test illegal packets with link-local addresses set as the source or destination.
 - a. Send packets with the source address set to a link-local address through the firewall both inbound and outbound.
 - b. Send packets with the destination address set to a link-local address through the firewall both inbound and outbound.
4. Test illegal packets with a loopback address set as the source.
 - a. Send a packet with the source address set to the ::1 loopback address through the firewall both inbound and outbound.
5. Test illegal packets with an IPv4-mapped address set as the source or destination.
 - a. Send a packet with the source address set to an IPv4-mapped address outbound and inbound through the firewall.
 - b. Send a packet with the destination address set to an IPv4-mapped address outbound and inbound through the firewall.
6. Test illegal packets with an IPv4-compatible address set as the source or destination.
 - a. Send a packet with the source address set to an IPv4-compatible address outbound and inbound through the firewall.

- b. Send a packet with the destination address set to an IPv4-compatible address outbound and inbound through the firewall.

Results:

1. Verify that no illegal packets pass through the firewall in either direction.

2.1.2: Extension Header Types

2.1.2.1: Selectively Block IPv6 packets based on Next Header values.

Requirement:

The firewall MUST selectively block any IPv6 packet based on its Next Header value.

Test Configuration:

Inbound:

Allow All:

Deny:

| Source Address | Destination Address | Next Header |
|----------------|---------------------|-------------|
| Public net | Private net | Set Value |

Outbound:

Allow All:

Deny:

| Source Address | Destination Address | Next Header |
|----------------|---------------------|-------------|
| Private net | Public net | Set Value |

An individual drop or deny rule must be added for each individual Next Header being tested.

Procedure:

1. Send packets with all possible next header values inbound through the product.
2. Send packets with all possible next header values outbound through the product.

Results:

1. Verify that the access control rules operate as expected.

2.1.2.2: Type 0 Routing Headers

Requirement:

The firewall MUST have the ability to selectively block type 0 routing headers.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public net | Private net | All | All | All |

Deny:

| Source Address | Destination Address | Protocol |
|----------------|---------------------|----------------|
| Public net | Private net | Type 0 Routing |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private net | Public net | All | All | All |

Deny:

| Source Address | Destination Address | Protocol |
|----------------|---------------------|----------------|
| Private net | Public net | Type 0 Routing |

Procedure:

1. Send a packet with a type 0 routing header inbound through the firewall.
2. Send a packet with a type 0 routing header outbound through the firewall.

Results:

1. Verify that the firewall blocks both of the packets sent.

2.1.3.1: TCP/UDP ports

Requirement:

The firewall MUST selectively allow/block any IPv6 packet sent inbound or outbound through its interfaces based on either the source or destination port in the packet sent.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public6-a | Private6-a | Any | 80 | TCP |
| Public6-a | Private6-a | Any | 53 | UDP |

Deny:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public6-a | Private6-a | 9001 | 80 | TCP |
| Public6-a | Private6-a | 9001 | 53 | UDP |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private6-b | Public6-b | Any | 80 | TCP |
| Private6-b | Public6-b | Any | 53 | UDP |

Deny:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private6-b | Public6-b | 9001 | 80 | TCP |
| Private6-b | Public6-b | 9001 | 53 | UDP |

Procedure:

1. Send packets inbound that trigger each configured inbound access control rule.
2. Send packets outbound that trigger each configured outbound access control rule.

Results:

1. Verify the allowed legitimate IPv6 traffic is properly passed by the firewall.
2. Verify that the firewall is blocking packets based on the source port set, in this example 9001, of each denied packet that attempts to traverse the firewall.

2.1.3.2. ICMPv6

Requirement:

The firewall MUST selectively allow/block ICMPv6 traffic by type and code.

Test Configuration:

The firewall will be configured with the following ICMPv6 policy:

Inbound:

Allow:

| Type | Code | Protocol |
|------|-------------------------------|----------|
| 1 | 0-6 (Destination Unreachable) | ICMP |
| 2 | 0 (Packet Too Big) | ICMP |
| 3 | 0,1 (Time Exceeded) | ICMP |
| 4 | 0-2 (Parameter Problem) | ICMP |
| 129 | 0 (Echo Response) | ICMP |

Deny:

All other type/code combinations.

Outbound:

Allow:

| Type | Code | Protocol |
|------|-------------------------------|----------|
| 1 | 0-6 (Destination Unreachable) | ICMP |
| 2 | 0 (Packet Too Big) | ICMP |
| 3 | 0 (Time Exceeded) | ICMP |
| 4 | 1,2 (Parameter Problem) | ICMP |
| 128 | 128 (Echo Request) | ICMP |

Deny:

All other type/code combinations.

Procedure:

1. Generate and send allowed legitimate ICMPv6 messages through the firewall in both inbound and outbound directions per the policy above.
2. Generate and send denied but otherwise properly formed ICMPv6 messages through the firewall in both inbound and outbound directions per the policy above. Be sure to send not only denied types, but allowed types with denied codes.

Results:

1. Verify the allowed legitimate ICMPv6 messages are properly forwarded to the intended destination.
2. Verify the denied ICMPv6 messages are dropped and no response is generated by the firewall.

2.1.4.1: Implicit Deny Policy

Requirement:

The firewall MUST block any traffic that has not been explicitly allowed.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public net | Private net | Any | 80 | TCP |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private net | Public net | Any | 80 | TCP |

Procedure:

1. Send traffic that violates the allowed access rules inbound through the firewall.
2. Send traffic that violates the allowed access rules outbound through the firewall.

Results:

1. Verify that denied traffic does not pass inbound or outbound through the firewall.

2.2: Asymmetrical blocking

2.2.1: Asymmetrical Controls.

Requirement:

The firewall MUST distinguish between internal and external networks and allow asymmetrical controls of traffic between these networks.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public6-a | Private-6a | Any | 80 | TCP |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private6-b | Public6-b | Any | 80 | TCP |

Procedure:

1. Send inbound TCP port 80 traffic from Public6-a to Private-6a.
2. Attempt to send outbound TCP port 80 traffic from Private-6a to Public6-a.
3. Send outbound TCP port 80 traffic from Private6-b to Public6-b.
4. Attempt to send inbound TCP port 80 traffic from Public6-b to Private6-b.

Results:

1. Verify that procedure step 1 allows TCP port 80 traffic.
2. Verify that procedure step 2 denies TCP port 80 traffic.
3. Verify that procedure step 3 allows TCP port 80 traffic.
4. Verify that procedure step 4 denies TCP port 80 traffic.

2.2.2: Allowing Connection Oriented Protocols.

Requirement:

The firewall MUST allow connection oriented protocols such as TCP to travel bi-directionally.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public net | Private net | Any | 80 | TCP |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private net | Public net | Any | 80 | TCP |

Procedure:

1. Establish an outbound TCP connection to a web server running on port 80.
2. Establish an inbound TCP connection to a web server running on port 80.

Results:

1. Verify that both of the inbound and outbound connection attempts were successful.

2.2.3: Selectively Blocking External Replies from the external network:

Requirement:

The firewall SHOULD be able to selectively block unsolicited replies from the external network.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public net | Private net | All | All | All |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private net | Public net | All | All | All |

Procedure:

1. Send an ICMP echo response inbound to a host.

Results:

1. Verify that the unsolicited ICMP echo response is blocked.

2.3: IPSec Traffic Handling:

2.3.1: ESP and AH Traffic Handling:

Requirement:

The firewall MUST selectively block ESP and AH traffic.

Test Configuration:

Inbound:

Deny:

| Source Address | Destination Address | Protocol |
|----------------|---------------------|----------|
| Any | Any | ESP [50] |
| Any | Any | AH [51] |

Allow:

| Source Address | Destination Address | Protocol |
|----------------|---------------------|----------|
| Any | Any | IPv6 |

Outbound:

Deny:

| Source Address | Destination Address | Protocol |
|----------------|---------------------|----------|
| Any | Any | ESP [50] |
| Any | Any | AH [51] |

Allow:

| Source Address | Destination Address | Protocol |
|----------------|---------------------|----------|
| Any | Any | IPv6 |

Procedure:

1. Send allowed legitimate traffic through the firewall in both inbound and outbound directions per the policy above.
2. Attempt to send legitimate ESP and AH traffic through the firewall in both inbound and outbound directions per the policy above.

Results:

1. Verify allowed traffic is passed.
2. Verify ESP and AH traffic is blocked.

2.3.2: Establishing Security Associations with IPsec.

Requirement:

The firewall MAY be capable of establishing Security Associations with IPsec hosts and other Security Gateways and meet IPsec Security, IKEv2, and Use of Cryptographic Algorithm requirements for a router as specified in the USGv6 Profile.

Note:

Not tested as part of the NPD test plan. Refer to the applicable Conformance and Interoperability Test Selections.

2.4: Performance under load:

2.4.1: Fail-Safe

Requirement:

When the firewall is suffering performance degradation due to overuse of resources it MUST fail in a manner that does not allow unauthorized access to itself or internal and external networks attached.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Public net | Private net | Any | 80 | TCP |

Outbound:

Allow:

| Source Address | Destination Address | Source Port | Destination Port | Protocol |
|----------------|---------------------|-------------|------------------|----------|
| Private net | Public net | Any | 80 | TCP |

Procedure:

1. Perform an operation on the firewall that will tax its resources
2. Attempt to send packets to the firewall that violate its configured security policy.
3. Attempt to send packets through the firewall inbound and outbound that violate its configured security policy.

Results:

1. Verify that the firewalls configured security policy was not violated.

2.5: Tunneled traffic handling:

Requirement:

The firewall MUST have the ability to handle all v4/v6 tunneling schemes, no matter how embedded, either by analyzing and applying the appropriate controls based on the encapsulated packet header, or by simply blocking all unanalyzed tunneled packets.

Test Configuration:

Inbound:

Allow:

| Source Address | Destination Address | Protocol |
|----------------|---------------------|----------|
| Public net | Private net | Any |

Outbound:

Allow:

| Source Address | Destination Address | Protocol |
|----------------|---------------------|----------|
| Private net | Public net | Any |

Procedure:

1. Place hardware on both the private and public networks configured with each possible tunnel configuration (6in4, 4in6, 6to4, and Teredo).
2. Enable tunnels.
3. If the firewall can analyze the traffic encapsulated in the tunnel attempt to apply access rules to it.
4. If the firewall can not analyze the traffic encapsulated in the tunnel, verify that it can be configured to drop any attempts to establish a tunnel.

Results:

1. Verify that the firewall can either drop any tunnels or apply access rules on any encapsulated traffic.

3. Application Firewalls

Default Test Configuration:

Initially configure the application firewall to enforce an appropriate security for the type of firewall per the vendor supplied documentation. Where necessary, specific configurations are detailed in the following test procedures.

3.1.1. Trust Barriers

Requirement:

The application firewall **MUST NOT** violate trust barriers by either:

- rewriting incoming untrusted data to appear trusted, or
- exposing information (e.g. internal network structures) to external untrusted networks.

Test Configuration:

Use the default configuration enforcing a security policy that selectively allows and blocks specific sessions and/or data.

Procedure:

1. Establish ongoing legitimate sessions to traverse the application firewall in accordance with the configured security policy.
2. Concurrently, attempt to pass incoming untrusted data through the application firewall that violates the configured security policy.

Results:

1. Verify legitimate sessions are established and traffic is passed.
2. Verify that untrusted data is blocked and not rewritten to appear as trusted data.
3. Verify that traffic passed does not expose internal information not explicitly permitted in the security policy.

3.2.1. Session traffic authorization

Requirement:

The application firewall **MUST** have the capability for controlled authorization for establishing sessions from the external network to internal hosts.

Test Configuration:

Use the default configuration enforcing a security policy that allows only authorized sessions and/or data.

Procedure:

1. Establish ongoing authorized sessions to traverse the application firewall in accordance with the configured security policy.
2. Concurrently, attempt to establish unauthorized sessions or pass unauthorized traffic from external networks to internal hosts through the application firewall.

Results:

1. Verify authorized sessions are established.
2. Verify unauthorized sessions are prevented and unauthorized traffic is blocked.

3.3.1. File filtering

Requirement:

The application firewall **MUST** have the capability to examine files (e.g. e-mail attachments) for malicious content and selectively block them. That is, the application firewall **MUST** provide sufficient means to block typical threat traffic.

Test Configuration:

Use the default configuration enforcing a security policy that examines files as they traverse the application firewall and blocks those with malicious content.

Procedure:

1. Establish ongoing legitimate sessions to traverse the application firewall passing appropriate legitimate files (no malicious content).
2. Concurrently, attempt to send files with malicious content through the application firewall.

Results:

1. Verify legitimate files are passed.
2. Verify that files with malicious content are blocked.

4. Intrusion Detection Systems

4.1: Known attack detection & Logging and alerts

Requirement:

The network IDS/IPS MUST be configurable to detect suspicious traffic based on known attack patterns. Refer to NIST USGv6 website for the set of vulnerabilities that are relevant to USG organizations.

Test Configuration:

IPv6 network traffic is passed through or to the network IDS/IPS by placing one or more monitoring segments of the device inline between switch ports configured as a trunk line or by connecting one or more monitoring ports on the device to a network tap or span port, respectively.

One or more systems capable of properly sending authentic attack traffic are connected to the test bed such that the attack traffic can be seen by the network IDS/IPS. One or more systems capable of properly emulating a vulnerable version of the relevant software are also connected to the test bed.

Procedure:

1. Examine the default policy on the network IDS/IPS
2. Configure and apply a policy on the network IDS/IPS that should detect traffic indicating an attempt to exploit a vulnerability relevant to USG organizations.
3. One at a time, generate or replay attack traffic that originates from a malicious host and targets a configured-to-be vulnerable system.
4. For each test case, make a note of the vulnerability that was targeted, the IP address of the attacker and victim, and the source or tool used to generate or replay the attack traffic.

Results:

1. Note whether the default policy on the device differs from the policy designed to detect attack traffic targeting vulnerabilities relevant to USG organizations.
2. If any policy changes were required, verify that each is reflected in the updated policy.
3. Document the response of the target system for each attack test case. Because the network IDS/IPS was placed in detection (and not in blocking) mode, the vulnerability should be exploited as expected.
4. Verify that a corresponding entry appears in the network IDS/IPS log for each attack test case.

4.2: Malformed packet detection

Requirement:

The network IDS/IPS MUST detect many different kinds of malformed and non-standard IPv6 frames.

Test Configuration:

IPv6 network traffic is passed through or to the network IDS/IPS by placing one or more monitoring segments of the device inline between switch ports configured as a trunk line or by connecting one or more monitoring ports on the device to a network tap or span port, respectively.

A device capable of generating and sending malformed IPv6 network traffic through the network IDS/IPS is connected to the test bed network. Either the same or a separate device configured and located appropriately to receive (or at least see) the sent IPv6 traffic is also connected to the test bed network.

Procedure:

1. Configure and apply a policy on the network IDS/IPS to detect malformed and/or non-standard IPv6 frames.
2. For each bullet below, create and send through the network IDS/IPS, one at a time, an IPv6 packet that is valid in all ways except:
 - a. the source address is set to ::1 (loopback address)
 - b. the destination address is set to 0 (unspecified address)
 - c. the source address is set to a multicast address
 - d. the destination address is set to ::1 (ensure sender actually puts packet on the wire and does not send the packet to itself)
 - e. the destination address is set to a reserved multicast address
 - f. it includes an undefined IPv6 option type
 - g. it includes an unassigned multicast address scope value
 - h. it includes an invalid value for the next header field
 - i. the next header is set to routing (43) and the routing header is set to hop-by-hop (00)
 - j. the next header is set to routing (43) and the routing header is also set to routing (43).
 - k. the next header is set to destination options (60) and the next header is set to routing (43).
 - l. it's an ICMPv6 Packet Too Big (message type 2) with MTU < 1280
 - m. it's an ICMPv6 packet of type 1 (destination unreachable) with code 2 (not assigned)
 - n. it's an ICMPv6 router solicitation packet (message type 133) with a code ≠ 0
 - o. it's an ICMPv6 router advertisement packet (message type 134) with a code ≠ 0
 - p. it's an ICMPv6 router advertisement packet (message type 134) with the reserved field ≠ 0
 - q. it's an ICMPv6 router advertisement packet (message type 134) with the reachable time field set > 1 hour
 - r. it's the final fragment with a zero offset (may have to precede this with a valid initial fragment packet)
3. Sniff at the destination or otherwise ensure that the network IDS/IPS was able to observe each sent IPv6 packet.

Results:

1. Record the time that each malformed and/or non-standard IPv6 frame is sent.
2. Monitor the log(s), a real time event viewer, or some other mechanism on the network IDS/IPS to confirm every malformed and/or non-standard IPv6 frame is detected by the network IDS/IPS.

4.3: Port-scanning detection

Requirement:

The network IDS/IPS MUST be capable of detecting typical port and host scans.

Test Configuration:

IPv6 network traffic is passed through or to the network IDS/IPS by placing one or more monitoring segments of the device inline between switch ports configured as a trunk line or by connecting one or more monitoring ports on the device to a network tap or span port, respectively.

A device capable of performing or properly emulating port and host scans is connected to the test bed network. A victim server or device capable of emulating a server is reachable by the scanning device. A variety of well known ports should be open, TCP and UDP. The network IDS/IPS is connected to the test bed network so that it can see the scanning activity.

Procedure:

1. Configure and apply a policy on the network IDS/IPS to detect port and host scans.
2. Run a TCP Connect scan from the scanning device against the victim device on the well known port numbers (reduce the delay between sent packets only if the delay between port scan probes is > 1 minute).
3. During the port scan, monitor the network traffic to ensure that the port scan traffic is passing through the network IDS/IPS.
4. During the port scan, monitor the network traffic at the scanning device to ensure that traffic being sent from the victim server is being received as and when expected.
5. Run a UDP scan from the scanning device against the victim device on the well known port numbers (reduce the delay between sent packets only if the delay between port scan probes is > 1 minute).
6. Repeat sniffing-related steps 3 and 4 for the UDP scan.

Results:

1. Record the start and end time for each port scan.
2. During each scan, periodically monitor the log(s), a real time event viewer, or some other mechanism on the network IDS/IPS to confirm that the scans are detected by the network IDS/IPS.

4.4: Tunneled traffic detection

Requirement:

When attack traffic is encapsulated using IPv6 tunneling, the network IDS/IPS **MUST** be able to detect suspicious traffic based on known attack patterns. Refer to NIST USGv6 website for the set of vulnerabilities that are relevant to USG organizations. When encapsulation is present that renders content inspection infeasible, the NPD **MUST** still be able to detect that encapsulation is present.

Test Configuration:

IPv6 network traffic is passed through or to the network IDS/IPS by placing one or more monitoring segments of the device inline between switch ports configured as a trunk line or by connecting one or more monitoring ports on the device to a network tap or span port, respectively.

A device capable of encapsulating and removing the encapsulation for 6in4, 4in6, 6to4, and Teredo tunneled traffic is connected to the test bed network. Either the same or a separate device configured and located appropriately to receive the encapsulated traffic is also connected to the test bed network. The network IDS/IPS under test is in a location where it can examine all tunneled traffic.

One or more systems capable of properly sending authentic attack traffic are connected to the test bed such that the attack traffic can be seen by the network IDS/IPS. One or more systems capable of properly emulating a vulnerable version of the relevant software are also connected to the test bed.

Procedure:

1. Configure and apply a policy on the network IDS/IPS should detect traffic indicating an attempt to exploit a vulnerability relevant to USG organizations.
2. Also configure and set the policy to detect all forms of IPv6 → IPv4 and IPv4 → IPv6 tunneling methods.
3. Configure the encapsulation-capable device(s) such that the traffic seen by the network IDS/IPS is 6in4 tunneled traffic
 - a. One at a time, generate or replay attack traffic that originates from a malicious host and targets a configured-to-be vulnerable system.
4. Repeat step 3.a. after re-configuring the encapsulation-capable device(s) such that the traffic seen by the network IDS/IPS is 4in6 tunneled traffic.
5. Repeat step 3.a. after re-configuring the encapsulation-capable device(s) such that the traffic seen by the network IDS/IPS is 6to4 tunneled traffic.
6. Repeat step 3.a. after re-configuring the encapsulation-capable device(s) such that the traffic seen by the network IDS/IPS is Teredo tunneled traffic.
7. Configure the encapsulation-capable device(s) such that the traffic seen by the network IDS/IPS is 4in6 encapsulated then encapsulated again using 6in4 and repeat step 3.a.

Results:

1. Document the response of the target system for each attack test case. Because the network IDS/IPS was placed in detection (and not in blocking) mode, the vulnerability should be exploited as expected.

2. For steps 3 through 6 above, monitor the log(s), a real time event viewer, or some other mechanism on the IDS/IPS after each successful attack for a corresponding indication that the attack traffic was detected.
3. For step 7 above, monitor the log(s), a real time event viewer, or some other mechanism on the IDS/IPS for an indication that either the attack traffic or the existence of encapsulation was detected.

4.5: Logging and alerts

Merged into 4.1

4.6: Performance under load, fail-safe

Requirement:

The network IDS/IPS MUST demonstrate when configured to do so that it notifies administrators (be it via an alert, a log message, etc.) when under severe load.

Test Configuration:

IPv6 network traffic is passed through or to the network IDS/IPS by placing one or more monitoring segments of the device inline between switch ports configured as a trunk line or by connecting one or more monitoring ports on the device to a network tap or span port, respectively. How many connections are needed depends on how many monitoring segments the network IDS/IPS has and what sort of throughput each segment can accommodate. Taken together, the configuration should be capable of delivering sufficient traffic to fully load the network IDS/IPS.

A traffic generation tool capable of filling the IDS/IPS' available bandwidth is connected to the test bed network such that the traffic can be seen by the network IDS/IPS.

Procedure:

1. Configure the policy on the network IDS/IPS to alert/log or otherwise indicate when the device begins to be stressed by the network traffic load.
2. Review the network IDS/IPS specification and determine the maximum throughput the device is expected to be capable of handling.
3. Configure the traffic generation tool to generate proper IPv6 network traffic with a reasonable mix of protocols.
4. Send traffic through the network IDS/IPS as follows:
 - a. Begin with about 30% of the expected maximum throughput from 2.
 - b. Slowly and incrementally raise the throughput until an indication of impending overload is observed or the network IDS/IPS fails.

Results:

1. Monitor the log(s), a real time event viewer, or whatever means by which the network IDS/IPS notifies an administrator of impending failure.

5. Intrusion Prevention Systems

5.1: Implement intrusion detection capabilities

Perform the following test cases from section 4:

- 4.2 Malformed packet detection
- 4.3 Port-scanning detection
- 4.4 Tunneled traffic detection
- 4.6 Performance under load, fail-safe

5.2: Stop or attenuate detected attacks & Logging and alerts

Requirement:

The network IPS **MUST** be able to detect and prevent attack traffic based on known attack patterns. Refer to NIST USGv6 website for the set of vulnerabilities that are relevant to USG organizations.

Test Configuration:

IPv6 network traffic is passed through or to the network IDS/IPS by placing one or more monitoring segments of the device inline between switch ports configured as a trunk line or by connecting one or more monitoring ports on the device to a network tap or span port, respectively.

One or more systems capable of properly sending authentic attack traffic are connected to the test bed such that the attack traffic can be seen by the network IPS. One or more systems capable of properly emulating a vulnerable version of the relevant software are also connected to the test bed. The victim system must be located relative to the network IPS in the test bed such that it is possible for countermeasures used by the network IPS to prevent the attack.

Procedure:

1. Examine the default policy on the network IPS.
2. Configure and apply a policy on the network IPS that should detect traffic indicating an attempt to exploit a vulnerability relevant to USG organizations.
3. One at a time, generate or replay attack traffic that originates from a malicious host and targets a configured-to-be vulnerable system.
4. For each test case, make a note of which vulnerability was targeted, the IP address of the attacker and intended victim, and the source or tool used to generate or replay the attack traffic.

Results:

1. Note whether the default policy on the device differs from the policy designed to block or attenuate attack traffic targeting relevant vulnerabilities.
2. If any policy changes were required, verify that each is reflected in the updated policy.

3. Document the response of the target system for each attack test case. Because the network IPS is in configured to prevent the attacks, verify that the vulnerability should not be exploited or the attack must somehow be attenuated.
4. Verify that a corresponding entry appears in the network IPS log(s) for each attack test case.