

## Firewall Filtering Requirements

### 1 Security Requirements

#### 1.1 Address Based Filtering

To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement a Stateful Traffic Filtering capability. That capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) and/or receiving (destination) applicable network traffic as well as on established connection information.

#### 1.2 Port Based Filtering

To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port identified in the network traffic as well as on established connection information.

#### 1.3 Stateful Inspection

Stateful packet inspection is used to aid in the performance of packet flow through the TOE. Rather than apply the ruleset against each packet that is processed at a TOE interface, the TOE will determine whether a packet belongs to an "approved" established connection. The minimum set of attributes that are used to determine whether a packet is part of an established session are mandated for TCP and UDP, and the ST author is allowed to add the ICMP protocol if they desire.

#### 1.4 Related Connection Filtering

This objective addresses the concept of "dynamic rule" creation, where due to the expected behavior of an application layer protocol, a new connection or path is created due to the creation of a connection that is allowed by the ruleset. The File Transfer Protocol is an example of such a protocol, where a data connection is created in response to an allowed command connection.

#### 1.5 System Monitoring

Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure firewall specific firewall rules to 'log' when network traffic is found to match the configured rule. As a result, matching a firewall rule configured to 'log' will result in informative event logs whenever a match occurs.

## 1.6 TOE Administration

To address the issues involved with a trusted means of administration of the Stateful Traffic Filtering capability this security objective, which originated in the NDPP, is extended as follows. Note that it is assumed that use of the functions indicated below is protected in accordance with the requirements in the NDPP. Compliant TOEs will provide the functions necessary for an administrator to configure the firewall rules that are enforced by the TOE.

## 1.7 Resource Utilisation Protection

As a stateful protocol, TCP consumes valuable resources on end-systems and any state-aware devices in the traffic path. This fact can be exploited to starve systems of resources resulting in a denial of service condition. Compliant TOE's will implement the ability to limit the number of incomplete TCP connections targeted at an end system on a protected network and this limit should be configurable by an administrator.

# 2 TOE Security Functional Requirements

## 2.1 FFW\_RUL\_EXT.1 Stateful Traffic Filtering

**FFW\_RUL\_EXT.1.1** The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE

**Application Note:** This element identifies the policy (Stateful Traffic Filtering) that is applied to the network packets that are processed at the TOE's interfaces. Every packet that is received at a TOE's interface either has the ruleset that expresses this policy applied, or it is determined that the packet belongs to an established connection. The remaining elements in this component provide the details of the policy.

It is important to note that the TOE, which also includes the underlying platform, cannot permit network packets to flow unless the ruleset contains a rule that permits the flow, or the packet is deemed to belong to an established connection that has been permitted to flow. This principle must hold true during TOE startup, and upon failures the TOE may encounter.

**FFW\_RUL\_EXT.1.2** The TSF shall process the following network traffic protocols:

- Internet Control Message Protocol version 4 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP).

**Application Note:** This element identifies the protocols and references the protocol definitions that serve to define the minimum set of network traffic can be interpreted by the TOE.

The RFC numbers referenced ensure that the TOE parses packets with a well known structure (e.g. headers, fields) and are compliant with the standards.

**FFW\_RUL\_EXT.1.3** The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - [Selection:Transport Layer Protocol, IPv6 Extension header type [assignment: list of fields in IPv6 extension header]]
- TCP
  - Source Port

- Destination Port
- UDP
  - Source Port
  - Destination Port
- and distinct interface.

**Application Note:** This element identifies the various attributes that are applicable when constructing rules to be enforced by this requirement – the applicable interface is a property of the TOE and the rest of the identified attributes are defined in the associated RFCs. Note that the ‘Transport Layer Protocol’ is the IPv4/IPv6 field that identifies the applicable protocol, such as TCP, UDP, ICMP, or GRE. IPv6 extension headers are defined in RFC 2460 and the ST author may specify which fields within each supported extension header, if any may be used as attributes in the construction of an inspection rule. Also, ‘Interface’ identified above is the external port where the applicable network traffic was received or will be sent.

**FFW\_RUL\_EXT.1.4** The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit or drop with the option to log the operation.

**Application Note:** This element defines the operations that can be associated with rules used to match network traffic. Note that the data to be logged is identified in the Security Audit requirements, Section 4.2.2.

**FFW\_RUL\_EXT.1.5** The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

**Application Note:** This element identifies where rules can be assigned. Specifically, a conforming TOE must be able to assign filtering rules to each of its available and distinct network interfaces that handle layer 3 and 4 network traffic. A distinct network interface can be physical or logical but it does not necessarily required to be visible from the network perspective (e.g. it does not need to have an IP address assigned to it).

Note that there could be a separate ruleset for each interface or alternately a shared ruleset that somehow associates rules with specific interfaces.

**FFW\_RUL\_EXT.1.6** The TSF shall:

- a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [selection: ICMP, no other protocols] based on the following network packet attributes:
  1. TCP: source and destination addresses, source and destination ports, sequence number, Flags;
  2. UDP: source and destination addresses, source and destination ports;

3. [selection: 'ICMP: source and destination addresses, [selection: type, code, [assignment: list of matching attributes]]', no other protocols].
- b) Remove existing traffic flows from the set of established traffic flows based on the following: [selection: session inactivity timeout, completion of the expected information flow].

**Application Note:** This element requires that the protocols be identified for which the TOE can determine and manage the state such that sessions can be established and are used to make traffic flow decisions as opposed to fully processing the configured rules. This element also requires that applicable attributes used to determine whether a network packet matches and established session are identified.

If ICMP is selected as a protocol the source and destination addresses are required to be considered when determining if a packet belongs to an established "connection". The type and code attributes may be used to provide a more robust capability in determining whether an ICMP packet is what is expected in an established connection flow. For example, one would not expect echo replies to be part of a flow if an echo request had not been received. The open assignment in the selection for ICMP attributes is left for implementations that may use IPv6 attributes.

Item b) in this element requires specification of how the firewall can determine that established information flows should be removed from the set of established information flows by observing events such as the termination of a TCP session initiated by either endpoint with FIN flags in the TCP packet. If protocols are handled differently, it is expected that the ST would identify those differences.

**[OPTIONAL] FFW\_RUL\_EXT.1.7** The TSF shall be able to process the following network protocols:

1. [selection: FTP, SIP, H.323: [assignment: other supported protocols], no other protocols],

in order to dynamically define rules or establish sessions allowing network traffic to flow.

- [selection: FTP: TCP data sessions in accordance with the FTP protocol as specified in RFC 959, [assignment: list of additionally supported protocols and the types of network traffic to be allowed based on those protocols], none].

**Application Note:** This element requires the specification of more complex protocols that require the firewall to allow network traffic flow even though an existing rule does not explicitly allow the flow. For example, the FTP protocol requires both a control connection and a data connection if a user is to transfer

files. While there are well-known ports involved, port 21 (control port on FTP server) and port 20 (data port on server in active mode), there are random ports > 1023 used on the client side. In passive mode, the FTP server may use a random port >1023 instead of port 20. The data connection is initiated by the client in passive mode, and imitated by the FTP server in active mode.

For these types of protocols, the establishment of a “new” connection is allowed, even though the ruleset may appear to deny it (e.g., since a rule cannot predict which random port will be used by the client or potentially the server, the default rule to deny may appear to apply). The TSF could create a dynamic rule that governs the traffic flow, or the TSF could implicitly allow the new connection to be established based on expectations of the protocol implementation as specified in the RFC or equivalent standard.

It is important to note that there is no expectation that any network packets be inspected beyond layer 4 (TCP/UDP). This requirement simply requires that the ST author specify the conditions under which a rule is dynamically inserted into the firewall to allow expected connections with unpredictable UDP/TCP ports to correctly be established.

If the ST Author includes additional protocols they must identify the RFC or equivalent standard that specifies the behavior of the protocol, as is done for FTP above.

**FFW\_RUL\_EXT.1.8** The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

The TSF shall drop and be capable of [selection: counting, logging]:

1. Packets which are invalid fragments;
2. Fragmented packets which cannot be re-assembled completely;
3. Packets where the source address of the network packet is defined as being on a broadcast network;
4. Packets where the source address of the network packet is defined as being on a multicast network;
5. The TSF shall reject and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;
6. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
7. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address “reserved for future use” (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
8. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an

- “unspecified address” or an address “reserved for future definition and use” (i.e. 2000::/3) as specified in RFC 3513 for IPv6;
9. The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
  10. [selection: [assignment: other default rules enforced by the TOE], no other rules].

**Application Note:** This element describes the minimum default rules that should always be applied. When packets are dropped based on the above rules, the TOE should be capable of logging or recording the drop action in a counter.

Item 1 and item 2 above express how the TOE processes fragmented packets. Item 1, introduces the notion of invalid fragments, and allows the ST author to define what constitutes an invalid fragment. An acceptable implementation could consider any fragmented packet as invalid. Another acceptable implementation could consider a fragmented packet that partially overlaps a previously received fragment as invalid. Item 2 ensures that the ruleset is only applied when a packet is reassembled to address the threat of fragmented packet attacks. Note that in item 1, the logging of an invalid fragment may not be able to include all the fields that are expected in a packet header due to pieces missing in the invalid fragment.

In item 5, the intent is that the “networks associated” with the network interface may be beyond the immediate subnet associated with the interface. For example, the network topology could include a router and a subsequent subnet “behind” the firewall interface. Strict Reverse Path Forwarding would be an acceptable implementation to determine if this is the case, where Loose RPF would not be acceptable. The use of Access Control Lists may be another example of an acceptable implementation that allows this default to be overridden.

Item 10, provides the ST author the ability to specify additional rules that are enforced (either with or without specification in the administrator defined ruleset). The type of rules specified here could include things such as filtering of Christmas tree packets, filtering of non-SYN packets not related to an existing connection, and filtering of split handshake connections. This element could also be used to express behavior that allows packet flow, such as an ICMP response due to a host being unreachable.

FFW\_RUL\_EXT.1.9 The TSF shall be capable of enforcing the following Stateful Traffic Filtering rules on all network traffic:

The TSF shall drop and be capable of [selection: counting, logging]:

1. The TSF shall reject and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;

2. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;

**Application Note:** This element describes default rules that the TOE should be configured to enforce. This differs from FFW\_RUL\_EXT.1.9 in that the rules defined above should be enforced by default.

**FFW\_RUL\_EXT.1.10** When FFW\_RUL\_EXT.1.6 or FFW\_RUL\_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW\_RUL\_EXT.1.5) in an administratively defined order.

**Application Note:** This element requires that an administrator is able to define the order in which configured filtering rules are processed for matches.

**FFW\_RUL\_EXT.1.11** When FFW\_RUL\_EXT.1.6 or FFW\_RUL\_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

**Application Note:** This element requires that, except when a packet is part of an established session, the behavior is always to deny network traffic when no rules apply and no other operations are required, though they are not necessarily prohibited.

**FFW\_RUL\_EXT.1.12** The TSF shall be capable of limiting an administratively configured number of half-open TCP connections:

1. where there is a common destination IP address and TCP port tuple;
2. and [selection;from a specific source IP address, any other scenarios]

in the event that the configured limit is reached, new connection attempts shall be dropped and capable of being [selection: counted, logged]

To prevent logging system overload, log messages should be rate-limited.

**Application Note:** A half-open TCP connection is one that has not completed the full three-way handshake as defined in RFC 793. Incomplete TCP connections i.e. those that have completed the SYN and SYN-ACK portions of the three-way handshake consume valuable resources in end hosts and stateful traffic filtering devices in the traffic path and, in sufficient volume, can lead to a denial of service condition. To protect itself, and any targeted protected services, compliant TOEs shall be capable of limiting the number of half-open TCP connections targeted at a specific destination IP address and port number. Optionally, the ST author may also define additional methods of policing i.e. a maximum number of half-open connections for a specific client (i.e. common source IP address).



### 3 Evaluation Activities

#### 3.1 FFW\_RUL\_EXT.1.1

##### 3.1.1 TSS

The evaluator shall verify that the TSS provides a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process.

The evaluator shall verify that the TSS also include a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describe the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets.

##### 3.1.2 Guidance

The operational guidance associated with this requirement is assessed in the subsequent test assurance activities.

##### 3.1.3 Tests

**Test 1:** The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should sourced from the non-protected network and be directed at a host located on the protected network, with packet sniffers listening to see if any network traffic is allowed through.

Note: The remaining testing associated with application of the ruleset is addressed in the subsequent test assurance activities.

#### 3.2 FFW\_RUL\_EXT.1.2

##### 3.2.1 TSS

The evaluator shall verify that the TSS indicates that the following protocols are supported:

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP)

The evaluator shall verify that the TSS describes how conformance with the identified RFCs has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

### 3.2.2 Guidance

The evaluator shall verify that the operational guidance indicates that the following protocols are supported:

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP)

If the guidance describes other protocols that are processed by the TOE, it should be made clear that those protocols were not considered as part of the TOE evaluation unless explicitly identified as part of FFW\_RULE\_EXT.1.7.

### 3.2.3 Tests

The testing associated with this requirement is addressed in the subsequent test assurance activities.

## 3.3 FFW\_RUL\_EXT.1.3/FFW\_RUL\_EXT.1.4/FFW\_RUL\_EXT.1.5

### 3.3.1 TSS

The evaluator shall verify that the TSS describes a stateful packet filtering policy and the following attributes are identified as being configurable within stateful traffic filtering rules for the associated protocols:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address

- Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

The evaluator shall verify that each rule can identify the following actions: permit or drop with the option to log the operation. The evaluator shall verify that the TSS identifies all interface types subject to the stateful packet filtering policy and explains how rules are associated with distinct network interfaces.

### 3.3.2 Guidance

The evaluators shall verify that the operational guidance identifies the following attributes as being configurable within stateful traffic filtering rules for the associated protocols:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

The evaluator shall verify that the operational guidance indicates that each rule can identify the following actions: permit, drop, and optionally log.

The evaluator shall verify that the operational guidance explains how rules are associated with distinct network interfaces.

### 3.3.3 Tests

**Test 1:** The evaluator shall use the instructions in the operational guidance to test that stateful packet filter firewall rules can be created that permit, drop, and optionally log packets for each of the following attributes:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

**Test 2:** Repeat the test assurance activity above to ensure that stateful traffic filtering rules can be defined for each distinct network interface type supported by the TOE.

The following tests are performed using a subset of the protocol specific attributes defined in Table XXYY. For each protocol (ICMPv4, ICMPv6 etc.), the evaluator shall make a random selection of attributes from the table such that at minimum, the defined number of unique attributes is tested.

Test 1: The evaluator shall configure the TOE to permit and log a random selection of 20 ICMPv4 Type and Code taken from Table XXYY. The evaluator will generate packets matching each configured ICMPv4 Type and Code in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

Test 2: The evaluator shall configure the TOE to deny and log a random selection of 20 ICMPv4 Type and Code taken from Table XXYY. The evaluator will generate packets matching each configured ICMPv4 Type and Code in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

Test 3: The evaluator shall configure the TOE to permit and log a random selection of 15 ICMPv6 Type and Code taken from Table XXYY. The evaluator will generate packets matching each configured ICMPv6 Type and Code in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

Test 4: The evaluator shall configure the TOE to deny and log a random selection of 15 ICMPv6 Type and Code taken from Table XXYY. The evaluator will generate packets matching each configured ICMPv6 Type and Code in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

Test 5: The evaluator shall configure the TOE to permit and log a random selection of 30 IPv4 Transport Layer Protocols taken from Table XXYY in conjunction with a specific source address and specific destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

Test 6: The evaluator shall configure the TOE to permit all traffic except to deny and log a random selection of 30 IPv4 Transport Layer Protocols taken from Table XXYY in conjunction with a specific source address and specific destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

Test 7: The evaluator shall configure the TOE to permit and log a random selection of 45 IPv6 Transport Layer Protocols taken from Table XXYY in conjunction with a specific source address and specific destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

Test 8: The evaluator shall configure the TOE to permit all traffic except to deny and log a random selection of 45 IPv6 Transport Layer Protocols taken from Table XXYY in conjunction with a specific source address and specific destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

Test 9: The evaluator shall configure the TOE to permit and log TCP using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are

permitted (i.e., by capturing the packets after passing through the TOE) and logged.

Test 10: The evaluator shall configure the TOE to deny and log TCP using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

Test 11: The evaluator shall configure the TOE to permit and log UDP using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

Test 12: The evaluator shall configure the TOE to deny and log UDP using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

### 3.4 FFW\_RUL\_EXT.1.6

#### 3.4.1 TSS

The evaluator shall verify that the TSS identifies the protocols that support stateful session handling. The TSS shall identify TCP, UDP, and ICMP if selected by the ST author.

The evaluator shall verify that the TSS describes how stateful sessions are established (including handshake processing) and maintained.

The evaluator shall verify that for TCP, the TSS identifies and describes the use of the following attributes in session determination: source and destination addresses, source and destination ports, sequence number, and individual flags.

The evaluator shall verify that for UDP, the TSS identifies and describes the following attributes in session determination: source and destination addresses, source and destination ports.

The evaluator shall verify that for ICMP (if selected), the TSS identifies and describes the following attributes in session determination: source and destination addresses, other attributes chosen in FFW\_RUL\_EXT.1.6.

The evaluator shall verify that the TSS describes how established stateful sessions are removed. The TSS shall describe how connections are removed for each protocol based on normal completion and/or timeout conditions. The TSS shall also indicate when session removal becomes effective (e.g., before the next packet that might match the session is processed).

### 3.4.2 Guidance

The evaluator shall verify that the operational guidance describes stateful session behaviors. For example, a TOE might not log packets that are permitted as part of an existing session.

### 3.4.3 Tests

**Test 1:** The evaluator shall configure the TOE to permit and log TCP traffic. The evaluator shall initiate a TCP session. While the TCP session is being established, the evaluator shall introduce session establishment packets with incorrect flags to determine that the altered traffic is not accepted as part of the session (i.e., a log event is generated to show the ruleset was applied). After a TCP session is successfully established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports, sequence number, flags) one at a time in order to verify that the altered packets are not accepted as part of the established session.

**Test 2:** The evaluator shall terminate the TCP session established per Test 1 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

**Test 3:** The evaluator shall expire (i.e., reach timeout) the TCP session established per Test 1 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

**Test 4:** The evaluator shall configure the TOE to permit and log UDP traffic. The evaluator shall establish a UDP session. Once a UDP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports) one at a time in order to verify that the altered packets are not accepted as part of the established session.

**Test 5:** The evaluator shall expire (i.e., reach timeout) the UDP session established per Test 4 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

**Test 6:** If ICMP is selected, the evaluator shall configure the TOE to permit and log ICMP traffic. The evaluator shall establish a session for ICMP as defined in the TSS. Once an ICMP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, other

attributes chosen in FFW\_RUL\_EXT.1.6) one at a time in order to verify that the altered packets are not accepted as part of the established session.

**Test 7:** If applicable, the evaluator shall terminate the ICMP session established per Test 6 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

**Test 8:** The evaluator shall expire (i.e., reach timeout) the ICMP session established per Test 6 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

### 3.5 [OPTIONAL] FFW\_RUL\_EXT.1.7

#### 3.5.1 TSS

The evaluator shall verify that the TSS identifies the protocols that can cause the automatic creation of dynamic packet filtering rules. In some cases rather than creating dynamic rules, the TOE might establish stateful sessions to support some identified protocol behaviors.

The evaluator shall verify that the TSS explains the dynamic nature of session establishment and removal. The TSS also shall explain any logging ramifications.

The evaluator shall verify that for each of the protocols selected, the TSS explains the dynamic nature of session establishment and removal specific to the protocol.

#### 3.5.2 Guidance

The evaluator shall verify that the operational guidance describes dynamic session establishment capabilities.

The evaluator shall verify that the operational guidance describes the logging of dynamic sessions consistent with the TSS.

#### 3.5.3 Tests

**Test 1:** The evaluator shall define stateful traffic filtering rules to permit and log traffic for each of the supported protocols and drop and log TCP and UDP ports above 1024. Subsequently, the evaluator shall establish a connection for each of the selected protocols in order to ensure that it succeeds. The evaluator shall examine the generated logs to verify they are consistent with the operational guidance.

**Test 2:** Continuing from Test 1, the evaluator shall determine (e.g., using a packet sniffer) which port above 1024 opened by the control protocol, terminate the connection session, and then verify that TCP or UDP (depending on the



protocol selection) packets cannot be sent through the TOE using the same source and destination addresses and ports.

**Test 3:** For each additionally supported protocol, the evaluator shall repeat the procedure above for the protocol. In each case the evaluator must use the applicable RFC or standard in order to determine what range of ports to block in order to ensure the dynamic rules are created and effective.

### 3.6 FFW\_RUL\_EXT.1.8

#### 3.6.1 TSS

The evaluator shall verify that the TSS identifies the following as packets that will be automatically dropped and are counted or logged:

- 1 ~~Obi~~ Packets which are invalid fragments, including a description of what constitutes an invalid fragment
- 2 Fragments that cannot be completely re-assembled
- 3 Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface
- 4 Packets where the source address is defined as being on a broadcast network
- 5 Packets where the source address is defined as being on a multicast network
- 6 Packets where the source address is defined as being a loopback address
- 7 Packets where the source address is defined as being a reserved address as specified in RFC 1918 for IPv4, and RFC 3513 for IPv6
- 8 The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address “reserved for future use” (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
- 9 The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” (i.e. 2000::/3) as specified in RFC 3513 for IPv6;
- 10 Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

Other packets defined in FFW\_RUL\_EXT.1.8.

#### 3.6.2 Guidance

The evaluator shall verify that the operational guidance describes packets that are discarded and potentially logged by default. If applicable protocols are identified, their descriptions need to be consistent with the TSS. If logging is

configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

### 3.6.3 Tests

**Test 1:** The evaluator shall test each of the conditions for automatic packet rejection in turn. In each case, the TOE should be configured to allow all network traffic and the evaluator shall generate a packet or packet fragment that is to be rejected. The evaluator shall use packet captures to ensure that the unallowable packet or packet fragment is not passed through the TOE.

**Test 2:** For each of the cases above, the evaluator shall use any applicable guidance to enable dropped packet logging or counting. In each case above, the evaluator shall ensure that the rejected packet or packet fragment was recorded (either logged or an appropriate counter incremented).

## 3.7 FFW\_RUL\_EXT.1.9

### 3.7.1 TSS

The evaluator shall verify that the TSS explains how the following traffic can be dropped and counted or logged:

1. Packets where the source address is equal to the address of the network interface where the network packet was received
2. Packets where the source or destination address of the network packet is a link-local address

### 3.7.2 Guidance

The evaluator shall verify that the operational guidance provides guidance on how the TOE can be

### 3.7.3 Tests

**Test 1:** The evaluator shall configure the TOE to drop and log network traffic where the source address of the packet matches that of the TOE network interface upon which the traffic was received. The evaluator shall generate suitable network traffic to match the configured rule and verify that the traffic is dropped and a log message generated.

### 3.8 FFW\_RUL\_EXT.1.10

#### 3.8.1 TSS

The evaluator shall verify that the TSS describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of administrator defined and ordered ruleset.

#### 3.8.2 Guidance

The evaluator shall verify that the operational guidance describes how the order of stateful traffic filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

#### 3.8.3 Tests

**Test 1:** The evaluator shall devise two equal stateful traffic filtering rules with alternate operations – permit and drop. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.

**Test 2:** The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

### 3.9 FFW\_RUL\_EXT.1.11

#### 3.9.1 TSS

The evaluator shall verify that the TSS describes the process for applying stateful traffic filtering rules and also that the behavior (either by default, or as configured by the administrator) is to deny packets when there is no rule match unless another required conditions allows the network traffic (i.e., FFW\_RUL\_EXT.1.6 or FFW\_RUL\_EXT.1.7).

#### 3.9.2 Guidance

The evaluator shall verify that the operational guidance describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the operational guidance provides the appropriate instructions to configure the behavior to deny packets with no matching rules.

#### 3.9.3 Tests

**Test 1:** The evaluator shall configure the TOE with no ICMPv4 rules. The evaluator will generate 20 packets with a random selection of the ICMPv4 Type and Code (from those defined in Table XXYY) in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE).

**Test 2:** The evaluator shall configure the TOE with no ICMPv6 rules. The evaluator will generate 15 packets with a random selection of the ICMPv6 Type and Code (from those defined in Table XXYY) in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE).

**Test 3:** The evaluator shall ensure that the TOE has no configured IPv4 rules. The evaluator will generate IPv4 packets with randomised source and destination addresses within the non-protected and protected network ranges. The evaluator shall verify that all IPv4 traffic is denied. (i.e., by capturing no applicable packets passing through the TOE).

**Test 4:** The evaluator shall ensure that the TOE has no configured IPv6 rules. The evaluator will generate IPv6 packets with randomised source and destination addresses within the non-protected and protected network ranges. The evaluator shall verify that all IPv6 traffic is denied. (i.e., by capturing no applicable packets passing through the TOE).

**Test 5:** The evaluator shall configure the TOE to permit and log IPv4 traffic with a specific transport layer protocol as defined in table XXYY. The evaluator shall generate IPv4 packets with randomised source and destination IPv4 addresses and a random selection of 30 IPv4 transport layer protocols that do not match the configured permit rule. The evaluator shall verify that all IPv4 traffic is denied. (i.e., by capturing no applicable packets passing through the TOE).

**Test 6:** The evaluator shall configure the TOE to permit and log IPv6 traffic with a specific transport layer protocol as defined in table XXYY. The evaluator shall generate IPv6 packets with randomised source and destination IPv6 addresses and a random selection of 45 IPv6 transport layer protocols that do not match the configured permit rule. The evaluator shall verify that all IPv6 traffic is denied. (i.e., by capturing no applicable packets passing through the TOE).

**Test 7:** The evaluator shall configure the TOE to permit and log TCP using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets that do not match the configured source and destination TCP ports in order to ensure that they are denied (i.e., by capturing the packets after passing through the TOE) and logged.

**Test 8:** The evaluator shall configure the TOE to permit and log UDP using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets that do not match the configured source and destination UDP ports in order to ensure that they are denied (i.e., by capturing the packets after passing through the TOE) and logged.

## 3.10 FFW\_RUL\_EXT.1.12

### 3.10.1 TSS

The evaluator shall verify that the TSS describes how the TOE tracks and maintains information relating to the number of half-open TCP connections. The TSS should identify how the TOE behaves when the administratively defined limit is reached and should describe under what circumstances stale half-open connections are removed (e.g. after a timer expires).

### 3.10.2 Guidance

The evaluator shall verify that the operational guidance describes the behaviour of imposing TCP half-open connection limits and its default state if unconfigured. The evaluator shall verify that the guidance clearly indicates the conditions under which new connections will be dropped e.g. per-destination or per-client.

### 3.10.3 Tests

**Test 1:** The evaluator shall define a TCP half-open connection limit applicable to a specific target host on the TOE. The evaluator shall generate TCP SYN requests to pass through the TOE to the defined system using a randomised source IP address and common destination IP address and TCP port number. The number of SYN requests should exceed the TCP half-open threshold defined on the TOE. TCP SYN-ACK messages should not be acknowledged. The evaluator shall verify through packet capture that once the defined TCP half-open threshold has been reached, subsequent TCP SYN packets are not transmitted to the target system. The evaluator shall verify that when the configured threshold is reached that, depending upon the selection, either a log entry is generated or a counter is incremented.

**Test 2:** If selected, the evaluator shall follow Test 1 above but shall configure the TOE to apply a TCP half-open connection limit to apply per-client. The TCP SYN requests should be then sourced from a fixed IP address with a random destination IP address (from a range within the protected network subnet) and TCP port number. SYN messages should be acknowledged with a SYN-ACK but no further SYN should be generated by the client.