Supporting Document
Mandatory Technical Document

Evaluation Activities for Stateful Traffic Filter Firewalls cPP

September-2014

Version 0.1

CCDB-<Reference from CCDB, in the form 'YYYY-MM-nnn'>

Foreword

This is a supporting document, intended to complement the Common Criteria version 3 and the associated Common Evaluation Methodology for Information Technology Security Evaluation.

Supporting documents may be "Guidance Documents", that highlight specific approaches and application of the standard to areas where no mutual recognition of its application is required, and as such, are not of normative nature, or "Mandatory Technical Documents", whose application is mandatory for evaluations whose scope is covered by that of the supporting document. The usage of the latter class is not only mandatory, but certificates issued as a result of their application are recognized under the CCRA.

This supporting document has been developed by the Network International Technical Community (NDFW-iTC) and is designed to be used to support the evaluations of products against the cPPs identified in section 1.1.

Technical Editor: Network International Technical Community (NDFW-iTC)

Document history:

V0.1, 5 September 2014 (Initial release for public review)

General Purpose: See section 1.1.

Field of special use: This Supporting Document applies to the evaluation of TOEs claiming conformance with the collaborative Protection Profile for Stateful Traffic Filter Firewalls [FWcPP].

Acknowledgements:

This Supporting Document was developed by the Network international Technical Community with representatives from industry, Government agencies, Common Criteria Test Laboratories, and members of academia.

Table of Contents

1 Introduction

1.1 Technology Area and Scope of Supporting Document

1 This Supporting Document defines the Evaluation Activities associated with the collaborative Protection Profile for Stateful Traffic Filter Firewalls [FWcPP]. Note that [FWcPP] also requires the use of the Evaluation Activities for network devices described in [ND-SD]. This Supporting Document defines only the additional activities for [FWcPP], over and above those in [ND-SD].

2 In addition to defining Evaluation Activities for the benefit of evaluators, the definitions in this Supporting Document aim to provide a common understanding for developers, evaluators and users as to what aspects of the TOE are tested in an evaluation against the associated cPPs, and to what depth the testing is carried out.

3 This Supporting Document is mandatory for evaluations of products that claim conformance to any of the following cPP(s):

a) collaborative Protection Profile for Stateful Traffic Filter Firewalls [FWcPP].

4 Although Evaluation Activities are defined mainly for the evaluators to follow, the definitions in this Supporting Document aim to provide a common understanding for developers, evaluators and users as to what aspects of the TOE are tested in an evaluation against the associated cPPs, and to what depth the testing is carried out. This common understanding in turn contributes to the goal of ensuring that evaluations against the cPP achieve comparable, transparent and repeatable results. In general the definition of Evaluation Activities will also help Developers to prepare for evaluation by identifying specific requirements for their TOE. The specific requirements in Evaluation Activities may in some cases clarify the meaning of SFRs, and may identify particular requirements for the content of Security Targets (especially the TOE Summary Specification), user guidance documentation, and possibly supplementary information (e.g. for entropy analysis or cryptographic key management architecture).

1.2 Structure of the Document

5 Evaluation Activities can be defined for both Security Functional Requirements and Security Assurance Requirements. These are defined in separate sections of this Supporting Document.

6 If any Evaluation Activity cannot be successfully completed in an evaluation then the overall verdict for the evaluation is a ‘fail’. In rare cases there may be acceptable reasons why an Evaluation Activity may be modified or deemed not applicable for a particular TOE, but this must be agreed with the Certification Body for the evaluation.

7 In general, if all Evaluation Activities (for both SFRs and SARs) are successfully completed in an evaluation then it would be expected that the overall verdict for the evaluation is a ‘pass’. To reach a ‘fail’ verdict when the Evaluation Activities have been successfully completed would require a specific justification from the evaluator as to why the Evaluation Activities were not sufficient for that TOE.

8 Similarly, at the more granular level of Assurance Components, if the Evaluation Activities for an Assurance Component and all of its related SFR Evaluation Activities are successfully completed in an evaluation then it would be expected that the verdict for the Assurance Component is a ‘pass’. To reach a ‘fail’ verdict for the Assurance Component when these Evaluation Activities have been successfully completed would require a specific justification from the evaluator as to why the Evaluation Activities were not sufficient for that TOE.

1.3 Glossary

9 For definitions of standard CC terminology see [CC] part 1.

10 **cPP** – collaborative Protection Profile

11 **CVE** – Common Vulnerabilities and Exposures (database)

12 **iTC** – International Technical Community

13 **SD** – Supporting Document

14 **Supplementary information –** information that is not necessarily included in the Security Target or operational guidance, and that may not necessarily be public. Examples of such information could be entropy analysis, or description of a cryptographic key management architecture used in (or in support of) the TOE. The requirement for any such supplementary information will be identified in the relevant cPP (see description in section 4).

2 Evaluation Activities for SFRs

2.1 Firewall (FFW)

2.1.1 FFW_RULEXT.1 Stateful Traffic Filtering

2.1.1.1 TSS

15 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.

16 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.

2.1.1.2 Operational Guidance

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

2.1.1.3 Tests

18 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 be sourced and be directed at a host, with packet sniffers listening to see if any network traffic is allowed through.

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

FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4

TSS

20 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

21 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.

Operational Guidance

22 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

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

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

Tests

25 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 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

26 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.

27 Note that these test activities should be performed in conjunction with those of FFW_RUL_EXT.1.9 where the effectiveness of the rules is tested. The test activities for FFW_RUL_EXT.1.9 define the protocol/attribute combinations required to be tested. If those combinations are configured manually, that will fulfil

the objective of these test activities, but if those combinations are configured otherwise (e.g., using automation), these test activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator.

FFW_RUL_EXT.1.5

TSS

28 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.

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

30 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.

31 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.

32 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.1.5.

33 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).

Operational Guidance

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

Tests

35 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.

36 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.

37 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.

38 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.

39 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.

40 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.1.5) one at a time in order to verify that the altered packets are not accepted as part of the established session.

41 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.

42 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.

[OPTIONAL] FFW_RUL_EXT.1.6

TSS

43 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.

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

45 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.

Operational Guidance

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

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

Tests

48 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.

49 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.

50 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.

FFW_RUL_EXT.1.7

TSS

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

1 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 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 Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

52 Other packets defined in FFW_RUL_EXT.1.7.

Operational Guidance

53 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.

Tests

54 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.

55 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).

FFW_RUL_EXT.1.8

TSS

56 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

Operational Guidance

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

Tests

58 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.

FFW_RUL_EXT.1.9

TSS

59 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.

Operational Guidance

60 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.

Tests

61 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.

62 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.

FFW_RUL_EXT.1.10

TSS

63 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.1.5 or FFW_RUL_EXT.1.6).

Operational Guidance

64 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.

Tests

65 For each attribute in FFW_RUL_EXT.1.2, the evaluator shall construct a test to demonstrate that the TOE can correctly compare the attribute from the packet header to the ruleset, and shall demonstrate both the permit and deny for each case. The evaluator shall check the log in each case to confirm that the relevant rule was applied. The evaluator shall record a packet capture for each test to demonstrate the correct TOE behaviour.

FFW_RUL_EXT.1.11

TSS

66 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).

Operational Guidance

67 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.

Tests

68 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.

69 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.

3 Evaluation Activities for SARs

70 No additional Evaluation Activities for SARs (over and above those in [ND-SD]) are defined here.

4 Required Supplementary Information

71 No additional Required Supplementary Information (over and above that in [ND-SD]) is defined here.

5 References

[CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model
CCMB-2012-09-001, Version 3.1 Revision 4, September 2012

[CC2] Common Criteria for Information Technology Security Evaluation,
Part 2: Security Functional Components,
CCMB-2012-09-002, Version 3.1 Revision 4, September 2012

[CC3] Common Criteria for Information Technology Security Evaluation,
Part 3: Security Assurance Components,
CCMB-2012-09-003, Version 3.1 Revision 4, September 2012

[CEM] Common Methodology for Information Technology Security Evaluation,
Evaluation Methodology,
CCMB-2012-09-004, Version 3.1, Revision 4, September 2012

[FWcPP] collaborative Protection Profile for Stateful Traffic Filter Firewalls,
Version 0.1, 5 September 2014

[NDcPP] collaborative Protection Profile for Network Devices,
Version 0.1, 5 September 2014

[ND-SD] Evaluation Activities for Network Device cPP,
Version 0.1, September 2014

[VAWP] Draft cPP Vulnerability Analysis Whitepaper
[Other details TBD]

A. Vulnerability Analysis

A.1 Introduction

72 As noted in [VAWP], while vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator follows a set of well-defined activities and documents his findings such that others can follow his arguments and come to the same conclusion as the evaluator in his report. While this does not guarantee that different evaluation facilities will identify exactly the same type of vulnerabilities or come to exactly the same conclusions, the approach defines the minimum level of analysis and the scope of that analysis, and provides schemes a measure of assurance that that minimum level of analysis is being performed by the evaluation facilities.

73 This supplemental guidance provides the information described in [VAWP] for the Stateful Traffic Filter Firewall cPP, with modifications specific to this technology type.

A.2 Additional Documentation

74 [VAWP] indicates that the iTC determines appropriate additional documentation, based on the technology type, that will be made available to the evaluation team by the TOE developer. This documentation is in addition to that called out in the cPP evaluation activities and other SARs.

75 For the TFFW cPP, the additional documentation will at a minimum include the list of software and hardware components that comprise the TOE. Hardware components apply to all systems claimed in the ST, and should identify at a minimum the network hardware and processors used by the TOE. Software components include the underlying operating environment/operating system, plus major components such as a web server, libraries such as protocol or cryptographic libraries, etc. This additional documentation is merely a list of the name and version number of the components, and will be used by the evaluators in formulating hypotheses during their analysis.

A.3 Sources of vulnerability information

76 The method to be used in the vulnerability analysis for cPPs as outlined in [VAWP] is based on the flaw hypothesis methodology, where the evaluation team hypothesizes flaws and then either proves or disproves those flaws. Flaws are drawn from four sources:

1. A list of flaw hypotheses applicable to the technology described by the cPP (in this case, a firewall) derived from Common Vulnerability Enumeration (CVE) or similar sources—there is a fixed set in the cPP/supplemental guidance that are agreed to by the iTC. Additionally,

this will be supplemented with CVEs that are directly applicable to the TOE or its identified components. The evaluators will also include in their assessment applicable CVEs that have been issued since the cPP was published;

2. A list of flaw hypotheses listed in the cPP/supplemental guidance that are derived from lessons learned specific to that technology and other iTC input (that might be derived from other open sources and vulnerability databases, for example); and
3. A list of flaw hypotheses derived from information available to the evaluators based on the SFRs and the baseline evidence provided by the vendor described in the cPP/supplemental guidance, also including referenced public resources.
4. A list of flaw hypotheses that are generated through the use of TC-defined tools (e.g., nmap, fuzz testers) and their application may also be included.

77 Appendix (TBD-1) contains the list of CVE entries to be considered for flaw hypotheses of type 1 above. In order to supplement this list, the evaluators shall also perform a search on CVEs that are more recent that the publication date of the cPP, and those that are specific to the TOE and its components as specified by the additional documentation mentioned above. Any duplicates—either in specific CVE, or the flaw hypothesis that is generated from the CVE—can be noted and removed from consideration by the evaluation team.

78 The search criteria to be used when searching CVEs published after the publication date of the cPP shall include:

- The term “firewall”
- The following protocols: TCP, UDP, IPv4, IPv6
- Any protocols not listed above supported (through an SFR) by the TOE

79 As part of type 1 flaw hypothesis generation for the specific components of the TOE, the evaluator shall also search the component manufacturer’s websites to determine if flaw hypotheses can be generated on this basis (for instance, if security patches have been released for the version of the component being evaluated, the subject of those patches may form the basis for a flaw hypothesis).

80 Appendix (TBD-2) contains the list of flaw hypothesis generated by the iTC for this cPP.

81 With respect to type 3 flaws, the evaluator is free to formulate flaws that are based on information presented by the product (through on-line help, product documentation and user guides, etc.) and product behaviour during the (functional) testing activities. The evaluator is also free to formulate flaws that are based on material that is not part of the baseline evidence (e.g., information gleaned from an Internet mailing list, or reading interface documentation on interfaces not included in the set provided by the

developer), although such activities have the potential to vary significantly based upon the product and evaluation facility performing the analysis.

82 The evaluator shall perform the following activities to generate type 4 flaw hypotheses:

- Fuzz testing
  - Examine effects of sending:
    - mutated packets carrying each 'Type' and 'Code' value that is undefined in the relevant RFC for each of ICMPv4 (RFC 792) and ICMPv6 (RFC 4443)
    - mutated packets carrying each 'Transport Layer Protocol' value that is undefined in the respective RFC for each of IPv4 (RFC 791) and IPv6 (RFC 2460).

    Since none of these packets will match a rule, or belong to an allowed session, the packets should be dropped. The evaluator shall ensure the firewall does not allow these packets to flow through the TOE.
  - Mutation fuzz testing of the remaining fields in the required protocol headers. This testing requires sending mutations of well-formed packets that have both carefully chosen and random values inserted into each header field in turn. The carefully chosen values should include semantically significant values that can be determined from the type of the data that the field represents, such as values indicating positive and negative integers, boundary conditions, invalid binary combinations (e.g. for flag sets with dependencies between bits), and missing start or end values. Randomly chosen values can also lead to the device entering an insecure state.
  - Various open source and commercial penetration tools are potential sources of testing methodologies.

A.4 Process for Evaluator Vulnerability Analysis

83 As flaw hypotheses are generated from the activities described above, the evaluation team will attempt to prove or disprove the hypotheses. This process, as outlined in the [VAWP], is as follows.

84 The evaluator will refine each flaw hypothesis for the TOE and attempt to disprove it using the information provided by the developer or through penetration testing. During this process, the evaluator is free to interact with the developer without consulting the Scheme to determine if the flaw exists, including requests to the developer for additional evidence (e.g., detailed design information, consultation with engineering staff); however, the Scheme should be copied on all of these requests. Should the developer

object to the information being requested as being not compatible with the overall level of the evaluation activity/cPP and cannot provide evidence otherwise that the flaw is disproved, the evaluator prepares an appropriate set of materials as follows: the source documents used in formulating the hypothesis, and why it represents a potential compromise against a specific TOE function; an argument why the flaw hypothesis could not be proven or disproved by the evidence provided so far; and the type of information required to investigate the flaw hypothesis further. The Scheme will then either approve or disapprove the request for additional information. If approved, the developer provides the requested evidence to disprove the flaw hypothesis (or, of course, acknowledge the flaw).

85 For each hypothesis, the evaluator will note whether the flaw hypothesis has been successfully disproved, successfully proven to have identified a flaw, or requires further investigation to be performed as part of the penetration testing effort. Again this can be dealt with in terms of meetings or written charts. It is important to have the results documented.

86 Should a flaw be found (either through the developer agreeing with the documentation analysis, or through the penetration effort), the evaluator will report these flaws to the vendor. All confirmed flaws should be addressed by the developer, and the resolution should be agreed to by the evaluator and noted as part of the evaluation report.

A.5 Reporting

87 The evaluators shall produce two reports on the testing effort; one that is public-facing (that is, included in the non-proprietary evaluation report) and one that is delivered to the overseeing Scheme.

88 The public-facing report is just a statement that the lab has examined the CVEs applicable to the product and those specified in the cPP (this encompasses hypotheses of types 1 and 2 mentioned above). No other information is provided in the report.

89 For the (internal) Scheme report, we suggest that the evaluation team must report all of the flaw hypotheses generated; all documentation used to generate the flaw hypotheses; and how each flaw hypothesis was resolved (this includes whether the original flaw hypothesis was confirmed or disproved). In identifying the documentation used in coming up with the flaw hypotheses, the evaluation team must characterize the documentation so that a reader can determine whether it is strictly required by the support documents/assurance activities (that is, it forms part of the baseline evidence), and the nature of the documentation (design information, developer engineering notebooks, etc.). At the conclusion of the evaluation, a set of interested Schemes (subject to negotiation between all parties concerned) may review this information and make a determination of the impacts to supporting documents for future evaluations against that cPP (for example, if a large number of the flaw hypotheses were generated based on a certain type of documentation, then additional documentation in this area may be required for future evaluations).

B. Firewall Equivalency Considerations

90 No additional Equivalency Considerations (over and above those in [ND-SD]) are defined here.