



**Supporting Document**  
**Mandatory Technical Document**

---

Evaluation Activities for Stateful Traffic  
Filter Firewalls cPP

February-2015

Version 1.0

CCDB-2015-01-002

# 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:**

V1.0, 27 February 2015 (published version)

V0.4, 26 January 2015 (incorporates changes due to comments received from CCDB review)

V0.3, 17 October 2014 (released version following public review, submitted for CCDB review)

V0.2, 13 October 2014 (internal draft in response to public review comments, for iTC review)

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**

<b>1</b>	<b>INTRODUCTION.....</b>	<b>5</b>
1.1	Technology Area and Scope of Supporting Document .....	5
1.2	Structure of the Document .....	5
1.3	Glossary .....	6
<b>2</b>	<b>EVALUATION ACTIVITIES FOR SFRS .....</b>	<b>7</b>
2.1	User Data Protection (FDP) .....	7
2.1.1	FDP_RIP.2 Full Residual Information Protection .....	7
2.2	Firewall (FFW).....	7
2.2.1	FFW_RUL_EXT.1 Stateful Traffic Filtering .....	7
2.2.2	FFW_RUL_EXT.2 Stateful Filtering for Dynamic Protocols .....	15
<b>3</b>	<b>EVALUATION ACTIVITIES FOR SARS.....</b>	<b>17</b>
<b>4</b>	<b>REQUIRED SUPPLEMENTARY INFORMATION .....</b>	<b>18</b>
<b>5</b>	<b>REFERENCES.....</b>	<b>19</b>
<b>A.</b>	<b>VULNERABILITY ANALYSIS .....</b>	<b>20</b>
A.1	Introduction .....	20
A.2	Additional Documentation.....	20
A.3	Sources of vulnerability information .....	20
A.4	Process for Evaluator Vulnerability Analysis.....	22
A.5	Reporting .....	22
A.6	CVE Entries for Flaw Hypotheses .....	22
A.7	Additional Flaw Hypotheses .....	22
A.8	iTC Activities – cPP and Supporting Document Maintenance.....	22
<b>B.</b>	<b>FIREWALL EQUIVALENCY CONSIDERATIONS .....</b>	<b>23</b>

# 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 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].
- 3 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

- 4 Evaluation Activities can be defined for both Security Functional Requirements and Security Assurance Requirements. These are defined in separate sections of this Supporting Document.
- 5 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.
- 6 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.

7

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

8

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

9

**cPP** – collaborative Protection Profile

10

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

11

**iTC** – International Technical Community

12

**SD** – Supporting Document

13

**Supplementary information** – information that is not necessarily included in the Security Target or guidance documentation, 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 User Data Protection (FDP)

#### 2.1.1 FDP\_RIP.2 Full Residual Information Protection

##### 2.1.1.1 TSS

14 “Resources” in the context of this requirement are network packets being sent through (as opposed to “to”, as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

### 2.2 Firewall (FFW)

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

##### 2.2.1.1 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.

##### 2.2.1.2 Guidance Documentation

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

### 2.2.1.3 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. The evaluator shall verify using a packet sniffer that none of the generated network traffic is permitted through the firewall during initialization.
- 19 Test 2: 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 be permitted by the ruleset should be sourced and be directed at a host. The evaluator shall verify using a packet sniffer that none of the generated network traffic is permitted through the firewall during initialization and is only permitted once initialization is complete.
- 20 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**

- 21 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

22 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.

***Guidance Documentation***

23 The evaluators shall verify that the guidance documentation 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

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

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

***Tests***

26 Test 1: The evaluator shall use the instructions in the guidance documentation 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

27 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.

28 Note that these test activities should be performed in conjunction with those of FFW\_RUL\_EXT.1.8 where the effectiveness of the rules is tested. The test activities for FFW\_RUL\_EXT.1.8 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**

29 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.

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

31 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.

- 32 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.
- 33 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.
- 34 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).

### ***Guidance Documentation***

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

### ***Tests***

- 36 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.
- 37 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.
- 38 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.
- 39 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.

- 40 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.
- 41 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.
- 42 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.
- 43 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.

## **FFW\_RUL\_EXT.1.6**

### **TSS**

- 44 The evaluator shall verify that the TSS identifies the following as packets that will be automatically dropped and are counted or logged:
- a) Packets which are invalid fragments, including a description of what constitutes an invalid fragment
  - b) Fragments that cannot be completely re-assembled
  - c) Packets where the source address is defined as being on a broadcast network
  - d) Packets where the source address is defined as being on a multicast network
  - e) Packets where the source address is defined as being a loopback address
  - f) 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;
  - g) 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. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
  - h) Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified
  - i) Other packets defined in FFW\_RUL\_EXT.1.6

### ***Guidance Documentation***

- 45 The evaluator shall verify that the guidance documentation 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***

- 46 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.
- 47 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.1.7**

### **TSS**

- 48 The evaluator shall verify that the TSS explains how the following traffic can be dropped and counted or logged:
- Packets where the source address is equal to the address of the network interface where the network packet was received
  - Packets where the source or destination address of the network packet is a link-local address
  - 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

### ***Guidance Documentation***

- 49 The evaluator shall verify that the guidance documentation describes how the TOE can be configured to implement the required rules. If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

### ***Tests***

- 50 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.

- 51 Test 2: The evaluator shall configure the TOE to drop and log network traffic where the source IP address of the packet fails to match the network reachability information of the interface to which it is targeted, e.g. if the TOE believes that network network 192.168.1.0/24 is reachable through interface 2, network traffic with a source address from the 192.168.1.0/24 network should be generated and sent to an interface other than interface 2. The evaluator shall verify that the network traffic is dropped and a log message generated.

## **FFW\_RUL\_EXT.1.8**

### **TSS**

- 52 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.

### ***Guidance Documentation***

- 53 The evaluator shall verify that the guidance documentation 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***

- 54 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.
- 55 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.1.9**

### **TSS**

- 56 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.2.1).

### ***Guidance Documentation***

- 57 The evaluator shall verify that the guidance documentation 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 guidance documentation provides

the appropriate instructions to configure the behavior to deny packets with no matching rules.

**Tests**

- 58 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.1.10**

**TSS**

- 59 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).

**Guidance Documentation**

- 60 The evaluator shall verify that the guidance documentation 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**

Test 1: The evaluator shall define a TCP half-open connection limit on the TOE. The evaluator shall generate TCP SYN requests to pass through the TOE to the target system using a randomised source IP address and common destination IP address. 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.

**2.2.2 FFW\_RUL\_EXT.2 Stateful Filtering for Dynamic Protocols**

**FFW\_RUL\_EXT.2.1**

**TSS**

- 61 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.

- 62      The evaluator shall verify that the TSS explains the dynamic nature of session establishment and removal. The TSS also shall explain any logging ramifications.
- 63      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.

### ***Guidance Documentation***

- 64      The evaluator shall verify that the guidance documentation describes dynamic session establishment capabilities.
- 65      The evaluator shall verify that the guidance documentation describes the logging of dynamic sessions consistent with the TSS.

### ***Tests***

- 66      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 guidance documentation.
- 67      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.
- 68      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      Evaluation Activities for SARs**

- 69        No additional Evaluation Activities for SARs (over and above those in [ND-  
SD]) are defined here.
- 70        However, additional details regarding Vulnerability Analysis activities for  
Stateful Traffic Filter Firewalls are given in Appendix A of the current  
document.

## **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 1.0, 27 February 2015
- [NDcPP] collaborative Protection Profile for Network Devices,  
Version 1.0, 27 February 2015
- [ND-SD] Evaluation Activities for Network Device cPP,  
Version 1.0, 27 February 2015
- [VAWP] Draft cPP Vulnerability Analysis Whitepaper  
Version 0.2 Draft

## A. Vulnerability Analysis

### A.1 Introduction

72 [ND-SD] contains the details of the vulnerability analysis process to be followed; that information is not repeated here. The additional information that is needed for vulnerability analysis for TOEs conforming to [FWcPP] is contained in the following sections.

### A.2 Additional Documentation

73 [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.

74 No additional documentation over and above that currently specified in [ND-SD] is required.

### A.3 Sources of vulnerability information

75 [ND-SD] identifies the 4 types flaws to be considered by the evaluation team. For each type, the following additional information is provided for TOEs conforming to [FWcPP].

#### A.3.1 Type 1 Hypotheses – Public Vulnerability Database-Based

76 Section A.6 of this supporting document contains the list of public vulnerability sources, and entries in those sources, to be considered for flaw hypotheses of type 1 to be considered for TOEs conforming to [FWcPP]. In order to supplement this list, the evaluators shall also perform a search on those sources in Section A.6 that are more recent than 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 a specific entry, or the flaw hypothesis that is generated from an entry from the same or a different source – can be noted and removed from consideration by the evaluation team.

77 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.

78 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).

## Vulnerability Analysis

### Type 2 Hypotheses – iTC Generated

79       Section **Error! Reference source not found.** contains the list of flaw hypothesis generated by the iTC for this cPP. Should the evaluators discover a type 3 or type 4 flaw that they believe should be considered in future versions of this cPP for inclusion in Section A.7, they will submit a non-proprietary write-up of the flaw for consideration by the iTC.

### Type 3 Hypotheses – Evaluation Team Generated

80       No additional guidance is provided over and above that contained in [ND-SD].

### Type 4 Hypotheses – Tool Generated

81       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).
  - 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 (i.e. testing is to include both carefully chosen and random insertion test cases). The original well-formed packets would be accepted as part of a normal existing communication stream and may still be accepted as valid packets when subject to the carefully chosen mutations (the individual packet alone would be valid although its contents may not be valid in the context of preceding and/or following packets), but will often not be valid packets when random values are inserted into fields. 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 may not result in well-formed packets, but are included nonetheless to see whether they can lead to the device entering an

insecure state. Any results that are unexpected (e.g., core dumps) are candidates for a flaw hypothesis.

82        The iTC has not identified a specific tool to be used in accomplishing the above flaw hypothesis generation activity, so any tool used by the evaluation team is acceptable. The evaluation team shall record in the test report the name, version, parameters, and results of all test tools used for this this activity.

#### **A.4 Process for Evaluator Vulnerability Analysis**

83        The process to be followed is described in [ND-SD].

#### **A.5 Reporting**

84        Reporting activities are described in [ND-SD].

#### **A.6 CVE Entries for Flaw Hypotheses**

85        No entries are currently defined for this list.

#### **A.7 Additional Flaw Hypotheses**

86        No entries are currently defined for this list.

#### **A.8 iTCT Activities – cPP and Supporting Document Maintenance**

87        As indicated in Appendix A of [ND-SD], the iTCT plays a key role in determining the scope of the vulnerability analysis with respect to what is publically reported. Section A.8 of [ND-SD] details the activities that the iTCT must perform in order to ensure that flaws to be investigated by the evaluation team are identified, are compatible with the overall level of assurance of a TOE conformant to this PP, and cover the areas of concern by the iTCT for this technology.

## B. Firewall Equivalency Considerations

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