

Benchmarking Methodology Working Group  
Internet-Draft  
Intended status: Informational  
Expires: October 18, 2021

B. Balarajah  
C. Rossenhoevel  
EANTC AG  
B. Monkman  
NetSecOPEN  
April 16, 2021

**Benchmarking Methodology for Network Security Device Performance**  
**draft-ietf-bmwg-ngfw-performance-08**

Abstract

This document provides benchmarking terminology and methodology for next-generation network security devices including next-generation firewalls (NGFW), next-generation intrusion detection and prevention systems (NGIDS/NGIPS) and unified threat management (UTM) implementations. This document aims to strongly improve the applicability, reproducibility, and transparency of benchmarks and to align the test methodology with today's increasingly complex layer 7 security centric network application use cases. The main areas covered in this document are test terminology, test configuration parameters, and benchmarking methodology for NGFW and NGIDS/NGIPS to start with.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 18, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Requirements</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Scope</a>	<a href="#">4</a>
<a href="#">4.</a>	<a href="#">Test Setup</a>	<a href="#">4</a>
<a href="#">4.1.</a>	<a href="#">Test Bed Configuration</a>	<a href="#">4</a>
<a href="#">4.2.</a>	<a href="#">DUT/SUT Configuration</a>	<a href="#">6</a>
<a href="#">4.2.1.</a>	<a href="#">Security Effectiveness Configuration</a>	<a href="#">12</a>
<a href="#">4.3.</a>	<a href="#">Test Equipment Configuration</a>	<a href="#">12</a>
<a href="#">4.3.1.</a>	<a href="#">Client Configuration</a>	<a href="#">12</a>
<a href="#">4.3.2.</a>	<a href="#">Backend Server Configuration</a>	<a href="#">15</a>
<a href="#">4.3.3.</a>	<a href="#">Traffic Flow Definition</a>	<a href="#">16</a>
<a href="#">4.3.4.</a>	<a href="#">Traffic Load Profile</a>	<a href="#">17</a>
<a href="#">5.</a>	<a href="#">Test Bed Considerations</a>	<a href="#">18</a>
<a href="#">6.</a>	<a href="#">Reporting</a>	<a href="#">19</a>
<a href="#">6.1.</a>	<a href="#">Introduction</a>	<a href="#">19</a>
<a href="#">6.2.</a>	<a href="#">Detailed Test Results</a>	<a href="#">20</a>
<a href="#">6.3.</a>	<a href="#">Benchmarks and Key Performance Indicators</a>	<a href="#">21</a>
<a href="#">7.</a>	<a href="#">Benchmarking Tests</a>	<a href="#">22</a>
<a href="#">7.1.</a>	<a href="#">Throughput Performance with Application Traffic Mix</a>	<a href="#">22</a>
<a href="#">7.1.1.</a>	<a href="#">Objective</a>	<a href="#">22</a>
<a href="#">7.1.2.</a>	<a href="#">Test Setup</a>	<a href="#">23</a>
<a href="#">7.1.3.</a>	<a href="#">Test Parameters</a>	<a href="#">23</a>
<a href="#">7.1.4.</a>	<a href="#">Test Procedures and Expected Results</a>	<a href="#">24</a>
<a href="#">7.2.</a>	<a href="#">TCP/HTTP Connections Per Second</a>	<a href="#">25</a>
<a href="#">7.2.1.</a>	<a href="#">Objective</a>	<a href="#">25</a>
<a href="#">7.2.2.</a>	<a href="#">Test Setup</a>	<a href="#">25</a>
<a href="#">7.2.3.</a>	<a href="#">Test Parameters</a>	<a href="#">26</a>
<a href="#">7.2.4.</a>	<a href="#">Test Procedures and Expected Results</a>	<a href="#">27</a>
<a href="#">7.3.</a>	<a href="#">HTTP Throughput</a>	<a href="#">28</a>
<a href="#">7.3.1.</a>	<a href="#">Objective</a>	<a href="#">28</a>
<a href="#">7.3.2.</a>	<a href="#">Test Setup</a>	<a href="#">28</a>
<a href="#">7.3.3.</a>	<a href="#">Test Parameters</a>	<a href="#">29</a>
<a href="#">7.3.4.</a>	<a href="#">Test Procedures and Expected Results</a>	<a href="#">31</a>
<a href="#">7.4.</a>	<a href="#">HTTP Transaction Latency</a>	<a href="#">32</a>
<a href="#">7.4.1.</a>	<a href="#">Objective</a>	<a href="#">32</a>
<a href="#">7.4.2.</a>	<a href="#">Test Setup</a>	<a href="#">32</a>



7.4.3.	Test Parameters . . . . .	32
7.4.4.	Test Procedures and Expected Results . . . . .	34
7.5.	Concurrent TCP/HTTP Connection Capacity . . . . .	35
7.5.1.	Objective . . . . .	35
7.5.2.	Test Setup . . . . .	35
7.5.3.	Test Parameters . . . . .	35
7.5.4.	Test Procedures and Expected Results . . . . .	37
7.6.	TCP/HTTPS Connections per Second . . . . .	38
7.6.1.	Objective . . . . .	38
7.6.2.	Test Setup . . . . .	38
7.6.3.	Test Parameters . . . . .	38
7.6.4.	Test Procedures and Expected Results . . . . .	40
7.7.	HTTPS Throughput . . . . .	41
7.7.1.	Objective . . . . .	41
7.7.2.	Test Setup . . . . .	41
7.7.3.	Test Parameters . . . . .	42
7.7.4.	Test Procedures and Expected Results . . . . .	44
7.8.	HTTPS Transaction Latency . . . . .	45
7.8.1.	Objective . . . . .	45
7.8.2.	Test Setup . . . . .	45
7.8.3.	Test Parameters . . . . .	45
7.8.4.	Test Procedures and Expected Results . . . . .	47
7.9.	Concurrent TCP/HTTPS Connection Capacity . . . . .	48
7.9.1.	Objective . . . . .	48
7.9.2.	Test Setup . . . . .	48
7.9.3.	Test Parameters . . . . .	48
7.9.4.	Test Procedures and Expected Results . . . . .	50
8.	IANA Considerations . . . . .	51
9.	Security Considerations . . . . .	51
10.	Contributors . . . . .	51
11.	Acknowledgements . . . . .	51
12.	References . . . . .	52
12.1.	Normative References . . . . .	52
12.2.	Informative References . . . . .	52
Appendix A.	Test Methodology - Security Effectiveness Evaluation	53
A.1.	Test Objective . . . . .	53
A.2.	Test Bed Setup . . . . .	53
A.3.	Test Parameters . . . . .	53
A.3.1.	DUT/SUT Configuration Parameters . . . . .	53
A.3.2.	Test Equipment Configuration Parameters . . . . .	54
A.4.	Test Results Validation Criteria . . . . .	54
A.5.	Measurement . . . . .	54
A.6.	Test Procedures and Expected Results . . . . .	55
A.6.1.	Step 1: Background Traffic . . . . .	55
A.6.2.	Step 2: CVE Emulation . . . . .	56
Appendix B.	DUT/SUT Classification . . . . .	56
Authors' Addresses	. . . . .	56



## **1. Introduction**

15 years have passed since IETF recommended test methodology and terminology for firewalls initially ([RFC3511]). The requirements for network security element performance and effectiveness have increased tremendously since then. Security function implementations have evolved to more advanced areas and have diversified into intrusion detection and prevention, threat management, analysis of encrypted traffic, etc. In an industry of growing importance, well-defined, and reproducible key performance indicators (KPIs) are increasingly needed as they enable fair and reasonable comparison of network security functions. All these reasons have led to the creation of a new next-generation network security device benchmarking document and this document obsoletes [RFC3511].

## **2. Requirements**

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

## **3. Scope**

This document provides testing terminology and testing methodology for modern and next-generation network security devices. It covers the validation of security effectiveness configurations of network security devices, followed by performance benchmark testing. This document focuses on advanced, realistic, and reproducible testing methods. Additionally, it describes test bed environments, test tool requirements, and test result formats.

## **4. Test Setup**

Test setup defined in this document is applicable to all benchmarking tests described in Section 7. The test setup MUST be contained within an Isolated Test Environment (see Section 3 of [RFC6815]).

### **4.1. Test Bed Configuration**

Test bed configuration MUST ensure that any performance implications that are discovered during the benchmark testing aren't due to the inherent physical network limitations such as the number of physical links and forwarding performance capabilities (throughput and latency) of the network devices in the test bed. For this reason, this document recommends avoiding external devices such as switches and routers in the test bed wherever possible.



In some deployment scenarios, the network security devices (Device Under Test/System Under Test) are connected to routers and switches which will reduce the number of entries in MAC or ARP tables of the Device Under Test/System Under Test (DUT/SUT). If MAC or ARP tables have many entries, this may impact the actual DUT/SUT performance due to MAC and ARP/ND (Neighbor Discovery) table lookup processes. This document also recommends using test equipment with the capability of emulating layer 3 routing functionality instead of adding external routers in the test bed.

The test bed setup Option 1 (Figure 1) is the RECOMMENDED test bed setup for the benchmarking test.

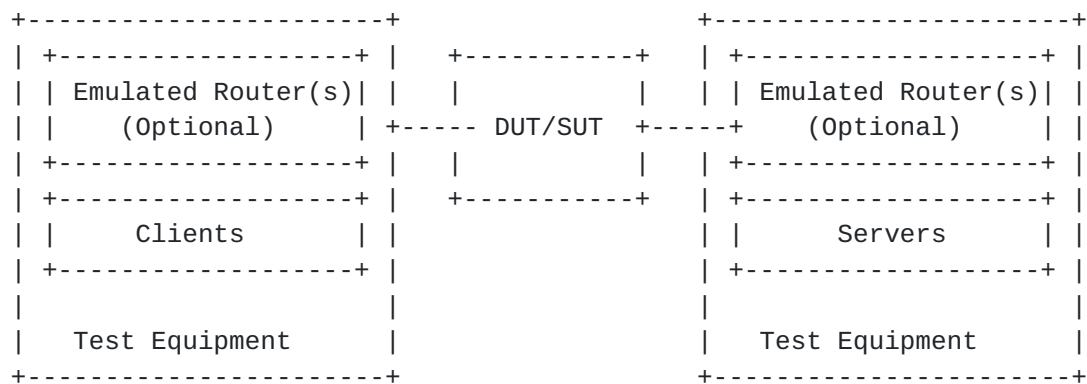


Figure 1: Test Bed Setup - Option 1

If the test equipment used is not capable of emulating layer 3 routing functionality or if the numbers of used ports are mismatched between test equipment and the DUT/SUT (need for a test equipment ports aggregation), the test setup can be configured as shown in Figure 2.







	NGFW	
DUT/SUT Features	RECOMMENDED	OPTIONAL
SSL Inspection	x	
IDS/IPS	x	
Anti-Spyware	x	
Anti-Virus	x	
Anti-Botnet	x	
Web Filtering		x
Data Loss Protection (DLP)		x
DDoS		x
Certificate Validation		x
Logging and Reporting	x	
Application Identification	x	

Table 1: NGFW Security Features



DUT/SUT Features	NGIDS/NGIPS	
	RECOMMENDED	OPTIONAL
SSL Inspection	x	
Anti-Malware	x	
Anti-Spyware	x	
Anti-Botnet	x	
Logging and Reporting	x	
Application Identification	x	
Deep Packet Inspection	x	
Anti-Evasion	x	

Table 2: NGIDS/NGIPS Security Features

The following table provides a brief description of the security features.

DUT/SUT Features	Description
SSL Inspection	DUT/SUT intercepts and decrypts inbound HTTPS traffic between servers and clients. Once the content inspection has been completed, DUT/SUT encrypts the HTTPS traffic with ciphers and keys used by the clients and servers.
IDS/IPS	DUT/SUT detects and blocks exploits targeting known and unknown vulnerabilities across the monitored network.
Anti-Malware	DUT/SUT detects and prevents the transmission of malicious executable code and any associated communications across the monitored network.



	This includes data exfiltration as well as command and control channels.
Anti-Spyware	Anti-Spyware is a subcategory of Anti Malware. Spyware transmits information without the user's knowledge or permission. DUT/SUT detects and block initial infection or transmission of data.
Anti-Botnet	DUT/SUT detects traffic to or from botnets.
Anti-Evasion	DUT/SUT detects and mitigates attacks that have been obfuscated in some manner.
Web Filtering	DUT/SUT detects and blocks malicious website including defined classifications of website across the monitored network.
DLP	DUT/SUT detects and blocks the transmission of Personally Identifiable Information (PII) and specific files across the monitored network.
Certificate Validation	DUT/SUT validates certificates used in encrypted communications across the monitored network.
Logging and Reporting	DUT/SUT logs and reports all traffic at the flow level across the monitored.
Application Identification	DUT/SUT detects known applications as defined within the traffic mix selected across the monitored network.

Table 3: Security Feature Description

In summary, a DUT/SUT SHOULD be configured as follows:

- o All RECOMMENDED security inspection enabled
- o Disposition of all flows of traffic are logged - Logging to an external device is permissible
- o Geographical location filtering and Application Identification and Control configured to be triggered based on a site or application from the defined traffic mix



In addition, a realistic number of access control rules (ACL) SHOULD be configured on the DUT/SUT where ACL's are configurable and also reasonable based on the deployment scenario. This document determines the number of access policy rules for four different classes of DUT/SUT; namely Extra Small (XS), Small (S), Medium (M) and Large (L). A sample DUT/SUT classification is described in [Appendix B](#).

The Access Control Rules (ACL) defined in Table 4 MUST be configured from top to bottom in the correct order as shown in the table. This is due to ACL types listed in specificity decreasing order, with "block" first, followed by "allow", representing typical ACL based security policy. The ACL entries SHOULD be configured with routable IP subnets by the DUT/SUT. (Note: There will be differences between how security vendors implement ACL decision making.) The configured ACL MUST NOT block the security and measurement traffic used for the benchmarking tests.



				+-----+				
					DUT/SUT			
					Classification			
					# Rules			
+-----+								
Rules Type	Match Criteria	Description	Action	XS	S	M	L	
+-----+								
Application layer	Application	Any application not included in the measurement traffic	block	5	10	20	50	
+-----+								
Transport layer	Src IP and TCP/UDP Dst ports	Any src IP subnet used and any dst ports not used in the measurement traffic	block	25	50	100	250	
+-----+								
IP layer	Src/Dst IP	Any src/dst IP subnet not used in the measurement traffic	block	25	50	100	250	
+-----+								
Application layer	Application	Half of the applications included in the measurement traffic (see the note below)	allow	10	10	10	10	
+-----+								
Transport layer	Src IP and TCP/UDP Dst ports	Half of the src IP used and any dst ports used in the measurement traffic (one rule per subnet)	allow	>1	>1	>1	>1	
+-----+								
IP layer	Src IP	The rest of the src IP subnet range used in the measurement traffic (one rule per subnet)	allow	>1	>1	>1	>1	
+-----+								

Table 4: DUT/SUT Access List



Note: If the half of applications included in the measurement traffic is less than 10, the missing number of ACL entries (dummy rules) can be configured for any application traffic not included in the measurement traffic.

#### **4.2.1. Security Effectiveness Configuration**

The Security features (defined in table 1 and 2) of the DUT/SUT MUST be configured effectively in such a way to detect, prevent, and report the defined security Vulnerability sets. This Section defines the selection of the security Vulnerability sets from Common Vulnerabilities and Exposures (CVE) list for the testing. The vulnerability set SHOULD reflect a minimum of 500 CVEs from no older than 10 calendar years to the current year. These CVEs SHOULD be selected with a focus on in-use software commonly found in business applications, with a Common Vulnerability Scoring System (CVSS) Severity of High (7-10).

This document is primarily focused on performance benchmarking. However, it is RECOMMENDED to validate the security features configuration of the DUT/SUT by evaluating the security effectiveness as a prerequisite for performance benchmarking tests defined in the [section 7](#). In case the Benchmarking tests are performed without evaluating security effectiveness, the test report MUST explain the implications of this. The methodology for evaluating Security effectiveness is defined in [Appendix A](#).

#### **4.3. Test Equipment Configuration**

In general, test equipment allows configuring parameters in different protocol layers. These parameters thereby influence the traffic flows which will be offered and impact performance measurements.

This section specifies common test equipment configuration parameters applicable for all benchmarking tests defined in [Section 7](#). Any benchmarking test specific parameters are described under the test setup section of each benchmarking test individually.

##### **4.3.1. Client Configuration**

This section specifies which parameters SHOULD be considered while configuring clients using test equipment. Also, this section specifies the RECOMMENDED values for certain parameters.



#### **4.3.1.1. TCP Stack Attributes**

The TCP stack SHOULD use a congestion control algorithm at client and server endpoints. The default IPv4 and IPv6 MSS segments size SHOULD be set to 1460 bytes and 1440 bytes respectively and a TX and RX initial receive windows of 64 KByte. Client initial congestion window SHOULD NOT exceed 10 times the MSS. Delayed ACKs are permitted and the maximum client delayed ACK SHOULD NOT exceed 10 times the MSS before a forced ACK. Up to three retries SHOULD be allowed before a timeout event is declared. All traffic MUST set the TCP PSH flag to high. The source port range SHOULD be in the range of 1024 - 65535. Internal timeout SHOULD be dynamically scalable per [RFC 793](#). The client SHOULD initiate and close TCP connections. The TCP connection MUST be initiated via a TCP three way handshake (SYN, SYN/ACK, ACK). and it MUST be closed via either a TCP three way close (FIN, FIN/ACK, ACK), or a TCP four way close (FIN, ACK, FIN, ACK).

#### **4.3.1.2. Client IP Address Space**

The sum of the client IP space SHOULD contain the following attributes.

- o The IP blocks SHOULD consist of multiple unique, discontinuous static address blocks.
- o A default gateway is permitted.
- o The IPv4 Type of Service (ToS) byte or IPv6 traffic class should be set to '00' or '000000' respectively.

The following equation can be used to define the total number of client IP addresses that will be configured on the test equipment.

Desired total number of client IP = Target throughput [Mbit/s] /  
Average throughput per IP address [Mbit/s]

As shown in the example list below, the value for "Average throughput per IP address" can be varied depending on the deployment and use case scenario.

(Option 1) DUT/SUT deployment scenario 1 : 6-7 Mbit/s per IP (e.g. 1,400-1,700 IPs per 10Gbit/s throughput)

(Option 2) DUT/SUT deployment scenario 2 : 0.1-0.2 Mbit/s per IP (e.g. 50,000-100,000 IPs per 10Gbit/s throughput)



Based on deployment and use case scenario, client IP addresses SHOULD be distributed between IPv4 and IPv6 type. The Following options can be considered for a selection of traffic mix ratio.

(Option 1) 100 % IPv4, no IPv6

(Option 2) 80 % IPv4, 20% IPv6

(Option 3) 50 % IPv4, 50% IPv6

(Option 4) 20 % IPv4, 80% IPv6

(Option 5) no IPv4, 100% IPv6

Note: The IANA has assigned IP address range for the testing purpose as described in [Section 8](#).

#### **[4.3.1.3](#). Emulated Web Browser Attributes**

The emulated web client contains attributes that will materially affect how traffic is loaded. The objective is to emulate modern, typical browser attributes to improve realism of the result set.

For HTTP traffic emulation, the emulated browser MUST negotiate HTTP 1.1. HTTP persistence MAY be enabled depending on the test scenario. The browser MAY open multiple TCP connections per Server endpoint IP at any time depending on how many sequential transactions are needed to be processed. Within the TCP connection multiple transactions MAY be processed if the emulated browser has available connections. The browser SHOULD advertise a User-Agent header. Headers MUST be sent uncompressed. The browser SHOULD enforce content length validation.

For encrypted traffic, the following attributes SHALL define the negotiated encryption parameters. The test clients MUST use TLS version 1.2 or higher. TLS record size MAY be optimized for the HTTPS response object size up to a record size of 16 KByte. If Server Name Indication (SNI) is required in the traffic mix profile, the client endpoint MUST send TLS Extension Server Name Indication (SNI) information when opening a security tunnel. Each client connection MUST perform a full handshake with server certificate and MUST NOT use session reuse or resumption.

The following TLS 1.2 supported ciphers and keys are RECOMMENDED to use for HTTPS based benchmarking tests defined in [Section 7](#).

1. ECHDE-ECDSA-AES128-GCM-SHA256 with Prime256v1 (Signature Hash Algorithm: ecdsa\_secp256r1\_sha256 and Supported group: secp256r1)



2. ECDHE-RSA-AES128-GCM-SHA256 with RSA 2048 (Signature Hash Algorithm: rsa\_pkcs1\_sha256 and Supported group: secp256)
3. ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 (Signature Hash Algorithm: ecdsa\_secp384r1\_sha384 and Supported group: secp521r1)
4. ECDHE-RSA-AES256-GCM-SHA384 with RSA 4096 (Signature Hash Algorithm: rsa\_pkcs1\_sha384 and Supported group: secp256)

Note: The above ciphers and keys were those commonly used enterprise grade encryption cipher suites for TLS 1.2. It is recognized that these will evolve over time. Individual certification bodies SHOULD use ciphers and keys that reflect evolving use cases. These choices MUST be documented in the resulting test reports with detailed information on the ciphers and keys used along with reasons for the choices.

[RFC8446] defines the following cipher suites for use with TLS 1.3.

1. TLS\_AES\_128\_GCM\_SHA256
2. TLS\_AES\_256\_GCM\_SHA384
3. TLS\_CHACHA20\_POLY1305\_SHA256
4. TLS\_AES\_128\_CCM\_SHA256
5. TLS\_AES\_128\_CCM\_8\_SHA256

#### **4.3.2. Backend Server Configuration**

This section specifies which parameters should be considered while configuring emulated backend servers using test equipment.

##### **4.3.2.1. TCP Stack Attributes**

The TCP stack on the server side SHOULD be configured similar to the client side configuration described in [Section 4.3.1.1](#). In addition, server initial congestion window MUST NOT exceed 10 times the MSS. Delayed ACKs are permitted and the maximum server delayed ACK MUST NOT exceed 10 times the MSS before a forced ACK.

##### **4.3.2.2. Server Endpoint IP Addressing**

The sum of the server IP space SHOULD contain the following attributes.



- o The server IP blocks SHOULD consist of unique, discontinuous static address blocks with one IP per Server Fully Qualified Domain Name (FQDN) endpoint per test port.
- o A default gateway is permitted. The IPv4 ToS byte and IPv6 traffic class bytes should be set to '00' and '000000' respectively.
- o The server IP addresses SHOULD be distributed between IPv4 and IPv6 with a ratio identical to the clients distribution ratio.

Note: The IANA has assigned IP address range for the testing purpose as described in [Section 8](#).

#### **4.3.2.3. HTTP / HTTPS Server Pool Endpoint Attributes**

The server pool for HTTP SHOULD listen on TCP port 80 and emulate HTTP version 1.1 with persistence. The Server MUST advertise server type in the Server response header [[RFC2616](#)]. For HTTPS server, TLS 1.2 or higher MUST be used with a maximum record size of 16 KByte and MUST NOT use ticket resumption or Session ID reuse. The server MUST listen on port TCP 443. The server SHALL serve a certificate to the client. The HTTPS server MUST check Host SNI information with the FQDN if the SNI is in use. Cipher suite and key size on the server side MUST be configured similar to the client side configuration described in [Section 4.3.1.3](#).

#### **4.3.3. Traffic Flow Definition**

This section describes the traffic pattern between client and server endpoints. At the beginning of the test, the server endpoint initializes and will be ready to accept connection states including initialization of the TCP stack as well as bound HTTP and HTTPS servers. When a client endpoint is needed, it will initialize and be given attributes such as a MAC and IP address. The behavior of the client is to sweep through the given server IP space, sequentially generating a recognizable service by the DUT. Thus, a balanced, mesh between client endpoints and server endpoints will be generated in a client IP and port server IP and port combination. Each client endpoint performs the same actions as other endpoints, with the difference being the source IP of the client endpoint and the target server IP pool. The client MUST use the server's IP address or Fully Qualified Domain Names (FQDN) in Host Headers [[RFC2616](#)].



#### **4.3.3.1. Description of Intra-Client Behavior**

Client endpoints are independent of other clients that are concurrently executing. When a client endpoint initiates traffic, this section describes how the client steps through different services. Once the test is initialized, the client endpoints randomly hold (perform no operation) for a few milliseconds to allow for better randomization of the start of client traffic. Each client will either open a new TCP connection or connect to a TCP persistence stack still open to that specific server. At any point that the traffic profile may require encryption, a TLS encryption tunnel will form presenting the URL or IP address request to the server. If using SNI, the server MUST then perform an SNI name check with the proposed FQDN compared to the domain embedded in the certificate. Only when correct, will the server process the HTTPS response object. The initial response object to the server is based on benchmarking tests described in [Section 7](#). Multiple additional sub-URLs (response objects on the service page) MAY be requested simultaneously. This MAY be to the same server IP as the initial URL. Each sub-object will also use a conical FQDN and URL path, as observed in the traffic mix used.

#### **4.3.4. Traffic Load Profile**

The loading of traffic is described in this section. The loading of a traffic load profile has five distinct phases: Init, ramp up, sustain, ramp down, and collection.

1. During the Init phase, test bed devices including the client and server endpoints should negotiate layer 2-3 connectivity such as MAC learning and ARP. Only after successful MAC learning or ARP/ND resolution SHALL the test iteration move to the next phase. No measurements are made in this phase. The minimum RECOMMEND time for Init phase is 5 seconds. During this phase, the emulated clients SHOULD NOT initiate any sessions with the DUT/SUT, in contrast, the emulated servers should be ready to accept requests from DUT/SUT or from emulated clients.
2. In the ramp up phase, the test equipment SHOULD start to generate the test traffic. It SHOULD use a set approximate number of unique client IP addresses actively to generate traffic. The traffic SHOULD ramp from zero to desired target objective. The target objective will be defined for each benchmarking test. The duration for the ramp up phase MUST be configured long enough, so that the test equipment does not overwhelm the DUT/SUT's stated performance metrics defined in [Section 6.3](#) namely; TCP Connections Per Second, Inspected Throughput, Concurrent TCP



Connections, and Application Transactions Per Second. No measurements are made in this phase.

3. Sustain phase starts when all required clients are active and operating at their desired load condition. In the sustain phase, the test equipment SHOULD continue generating traffic to constant target value for a constant number of active clients. The minimum RECOMMENDED time duration for sustain phase is 300 seconds. This is the phase where measurements occur.
4. In the ramp down/close phase, no new connections are established, and no measurements are made. The time duration for ramp up and ramp down phase SHOULD be the same.
5. The last phase is administrative and will occur when the test equipment merges and collates the report data.

## **5. Test Bed Considerations**

This section recommends steps to control the test environment and test equipment, specifically focusing on virtualized environments and virtualized test equipment.

1. Ensure that any ancillary switching or routing functions between the system under test and the test equipment do not limit the performance of the traffic generator. This is specifically important for virtualized components (vSwitches, vRouters).
2. Verify that the performance of the test equipment matches and reasonably exceeds the expected maximum performance of the system under test.
3. Assert that the test bed characteristics are stable during the entire test session. Several factors might influence stability specifically, for virtualized test beds. For example, additional workloads in a virtualized system, load balancing, and movement of virtual machines during the test, or simple issues such as additional heat created by high workloads leading to an emergency CPU performance reduction.

Test bed reference pre-tests help to ensure that the maximum desired traffic generator aspects such as throughput, transaction per second, connection per second, concurrent connection, and latency.

Test bed preparation may be performed either by configuring the DUT in the most trivial setup (fast forwarding) or without presence of the DUT.



## **6. Reporting**

This section describes how the final report should be formatted and presented. The final test report MAY have two major sections; Introduction and detailed test results sections.

### **6.1. Introduction**

The following attributes SHOULD be present in the introduction section of the test report.

1. The time and date of the execution of the test MUST be prominent.
2. Summary of test bed software and Hardware details

#### **A. DUT/SUT Hardware/Virtual Configuration**

- + This section SHOULD clearly identify the make and model of the DUT/SUT
- + The port interfaces, including speed and link information MUST be documented.
- + If the DUT/SUT is a Virtual Network Function (VNF), host (server) hardware and software details, interface acceleration type such as DPDK and SR-IOV used CPU cores, used RAM, and the resource sharing (e.g. Pinning details and NUMA Node) configuration MUST be documented. The virtual components such as Hypervisor, virtual switch version MUST be also documented.
- + Any additional hardware relevant to the DUT/SUT such as controllers MUST be documented

#### **B. DUT/SUT Software**

- + The operating system name MUST be documented
- + The version MUST be documented
- + The specific configuration MUST be documented

#### **C. DUT/SUT Enabled Features**

- + Configured DUT/SUT features (see Table 1 and Table 2) MUST be documented
- + Attributes of those featured MUST be documented



- + Any additional relevant information about features MUST be documented

D. Test equipment hardware and software

- + Test equipment vendor name
- + Hardware details including model number, interface type
- + Test equipment firmware and test application software version

E. Key test parameters

- + Used cipher suites and keys
- + IPv4 and IPv6 traffic distribution
- + Number of configured ACL

F. Details of application traffic mix used in the benchmarking test "Throughput Performance with Application Traffic Mix" ([Section 7.1](#))

- + Name of applications and layer 7 protocols
- + Percentage of emulated traffic for each application and layer 7 protocols
- + Percentage of encrypted traffic and used cipher suites and keys (The RECOMMENDED ciphers and keys are defined in [Section 4.3.1.3](#))
- + Used object sizes for each application and layer 7 protocols

3. Results Summary / Executive Summary

- A. Results SHOULD resemble a pyramid in how it is reported, with the introduction section documenting the summary of results in a prominent, easy to read block.

## **[6.2.](#) Detailed Test Results**

In the result section of the test report, the following attributes should be present for each benchmarking test.



- a. KPIs MUST be documented separately for each benchmarking test. The format of the KPI metrics should be presented as described in [Section 6.3](#).
- b. The next level of details SHOULD be graphs showing each of these metrics over the duration (sustain phase) of the test. This allows the user to see the measured performance stability changes over time.

### **6.3. Benchmarks and Key Performance Indicators**

This section lists key performance indicators (KPIs) for overall benchmarking tests. All KPIs MUST be measured during the sustain phase of the traffic load profile described in [Section 4.3.4](#). All KPIs MUST be measured from the result output of test equipment.

- o Concurrent TCP Connections  
The aggregate number of simultaneous connections between hosts across the DUT/SUT, or between hosts and the DUT/SUT (defined in [\[RFC2647\]](#)).
- o TCP Connections Per Second  
The average number of successfully established TCP connections per second between hosts across the DUT/SUT, or between hosts and the DUT/SUT. The TCP connection must be initiated via a TCP three way handshake (SYN, SYN/ACK, ACK). Then the TCP session data is sent. The TCP session MUST be closed via either a TCP three way close (FIN, FIN/ACK, ACK), or a TCP four way close (FIN, ACK, FIN, ACK), and not by a RST.
- o Application Transactions Per Second  
The average number of successfully completed transactions per second. For a particular transaction to be considered successful, all data must have been transferred in its entirety. In case of HTTP(S) transaction, it must have a valid status code, and the appropriate FIN, FIN/ACK sequence must have been completed.
- o TLS Handshake Rate  
The average number of successfully established TLS connections per second between hosts across the DUT/SUT, or between hosts and the DUT/SUT.
- o Inspected Throughput  
The number of bits per second of allowed traffic a network security device is able to transmit to the correct destination interface(s) in response to a specified offered load. The throughput benchmarking tests defined in [Section 7](#) SHOULD measure the average OSI model Layer 2 throughput value. This document



recommends presenting the throughput value in Gbit/s rounded to two places of precision with a more specific Kbit/s in parenthesis.

- o Time to First Byte (TTFB)  
TTFB is the elapsed time between the start of sending the TCP SYN packet from the client and the client receiving the first packet of application data from the server or DUT/SUT. The benchmarking tests HTTP Transaction Latency ([Section 7.4](#)) and HTTPS Transaction Latency ([Section 7.8](#)) measure the minimum, average and maximum TTFB. The value SHOULD be expressed in millisecond.
- o URL Response time / Time to Last Byte (TTLB)  
URL Response time / TTLB is the elapsed time between the start of sending the TCP SYN packet from the client and the client receiving the last packet of application data from the server or DUT/SUT. The benchmarking tests HTTP Transaction Latency ([Section 7.4](#)) and HTTP Transaction Latency ([Section 7.8](#)) measure the minimum, average and maximum TTLB. The value SHOULD be expressed in millisecond.

## **[7. Benchmarking Tests](#)**

### **[7.1. Throughput Performance with Application Traffic Mix](#)**

#### **[7.1.1. Objective](#)**

Using a relevant application traffic mix, determine the sustainable inspected throughput supported by the DUT/SUT.

Based on customer use case, users can choose the application traffic mix for this test. The details about the traffic mix MUST be documented in the report. At least the following traffic mix details MUST be documented and reported together with the test results:

Name of applications and layer 7 protocols

Percentage of emulated traffic for each application and layer 7 protocols

Percentage of encrypted traffic and used cipher suites and keys  
(The RECOMMENDED ciphers and keys are defined in [Section 4.3.1.3.](#))

Used object sizes for each application and layer 7 protocols



### **7.1.2. Test Setup**

Test bed setup MUST be configured as defined in [Section 4](#). Any benchmarking test specific test bed configuration changes MUST be documented.

### **7.1.3. Test Parameters**

In this section, the benchmarking test specific parameters SHOULD be defined.

#### **7.1.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented. In case the DUT is configured without SSL inspection feature, the test report MUST explain the implications of this to the relevant application traffic mix encrypted traffic.

#### **7.1.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be noted for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target inspected throughput: Aggregated line rate of interface(s) used in the DUT/SUT or the value defined based on requirement for a specific deployment scenario

Initial inspected throughput: 10% of the "Target inspected throughput"

One of the ciphers and keys defined in [Section 4.3.1.3](#) are RECOMMENDED to use for this benchmarking test.

#### **7.1.3.3. Traffic Profile**

Traffic profile: This test MUST be run with a relevant application traffic mix profile.



#### **7.1.3.4. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of total attempt transactions.
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections.

#### **7.1.3.5. Measurement**

Following KPI metrics MUST be reported for this benchmarking test:

Mandatory KPIs (benchmarks): Inspected Throughput, TTFB (minimum, average, and maximum), TTLB (minimum, average, and maximum) and Application Transactions Per Second

Note: TTLB MUST be reported along with the object size used in the traffic profile.

Optional KPIs: TCP Connections Per Second and TLS Handshake Rate

#### **7.1.4. Test Procedures and Expected Results**

The test procedures are designed to measure the inspected throughput performance of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IP types; IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution.

##### **7.1.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to generate test traffic at the "Initial inspected throughput" rate as described in the parameters [Section 7.1.3.2](#). The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#). The DUT/SUT SHOULD reach the "Initial inspected throughput" during the sustain phase. Measure all KPI as defined in [Section 7.1.3.5](#). The



measured KPIs during the sustain phase **MUST** meet the test results validation criteria "a" and "b" defined in [Section 7.1.3.4](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure **MUST NOT** be continued to step 2.

#### **[7.1.4.2](#). Step 2: Test Run with Target Objective**

Configure test equipment to generate traffic at the "Target inspected throughput" rate defined in the parameter table. The test equipment **SHOULD** follow the traffic load profile definition as described in [Section 4.3.4](#). The test equipment **SHOULD** start to measure and record all specified KPIs and the frequency of measurements **SHOULD** be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT is expected to reach the desired value of the target objective ("Target inspected throughput") in the sustain phase. Follow step 3, if the measured value does not meet the target value or does not fulfill the test results validation criteria.

#### **[7.1.4.3](#). Step 3: Test Iteration**

Determine the achievable average inspected throughput within the test results validation criteria. Final test iteration **MUST** be performed for the test duration defined in [Section 4.3.4](#).

### **[7.2](#). TCP/HTTP Connections Per Second**

#### **[7.2.1](#). Objective**

Using HTTP traffic, determine the sustainable TCP connection establishment rate supported by the DUT/SUT under different throughput load conditions.

To measure connections per second, test iterations **MUST** use the different fixed HTTP response object sizes (the different load conditions) defined in [Section 7.2.3.2](#).

#### **[7.2.2](#). Test Setup**

Test bed setup **SHOULD** be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. **MUST** be documented.



### **7.2.3. Test Parameters**

In this section, benchmarking test specific parameters SHOULD be defined.

#### **7.2.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented.

#### **7.2.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target connections per second: Initial value from product datasheet or the value defined based on requirement for a specific deployment scenario

Initial connections per second: 10% of "Target connections per second" (an optional parameter for documentation)

The client SHOULD negotiate HTTP 1.1 and close the connection with FIN immediately after completion of one transaction. In each test iteration, client MUST send GET command requesting a fixed HTTP response object size.

The RECOMMENDED response object sizes are 1, 2, 4, 16, and 64 KByte.

#### **7.2.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of total attempt transactions.



- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections.
- c. During the sustain phase, traffic should be forwarded at a constant rate.
- d. Concurrent TCP connections MUST be constant during steady state and any deviation of concurrent TCP connections SHOULD be less than 10%. This confirms the DUT opens and closes TCP connections almost at the same rate.

#### **7.2.3.4. Measurement**

TCP Connections Per Second MUST be reported for each test iteration (for each object size).

#### **7.2.4. Test Procedures and Expected Results**

The test procedure is designed to measure the TCP connections per second rate of the DUT/SUT at the sustaining period of the traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IP types; IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution.

##### **7.2.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure the traffic load profile of the test equipment to establish "initial connections per second" as defined in the parameters [Section 7.2.3.2](#). The traffic load profile SHOULD be defined as described in [Section 4.3.4](#).

The DUT/SUT SHOULD reach the "Initial connections per second" before the sustain phase. The measured KPIs during the sustain phase MUST meet the test results validation criteria a, b, c, and d defined in [Section 7.2.3.3](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure MUST NOT be continued to "Step 2".



#### **7.2.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish the target objective ("Target connections per second") defined in the parameters table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#).

During the ramp up and sustain phase of each test iteration, other KPIs such as inspected throughput, concurrent TCP connections and application transactions per second MUST NOT reach to the maximum value the DUT/SUT can support. The test results for specific test iterations SHOULD NOT be reported, if the above mentioned KPI (especially inspected throughput) reaches the maximum value. (Example: If the test iteration with 64 KByte of HTTP response object size reached the maximum inspected throughput limitation of the DUT, the test iteration MAY be interrupted and the result for 64 KByte SHOULD NOT be reported).

The test equipment SHOULD start to measure and record all specified KPIs and the frequency of measurements SHOULD be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT is expected to reach the desired value of the target objective ("Target connections per second") in the sustain phase. Follow step 3, if the measured value does not meet the target value or does not fulfill the test results validation criteria.

#### **7.2.4.3. Step 3: Test Iteration**

Determine the achievable TCP connections per second within the test results validation criteria.

### **7.3. HTTP Throughput**

#### **7.3.1. Objective**

Determine the sustainable inspected throughput of the DUT/SUT for HTTP transactions varying the HTTP response object size.

#### **7.3.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.



### **7.3.3. Test Parameters**

In this section, benchmarking test specific parameters SHOULD be defined.

#### **7.3.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented.

#### **7.3.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target inspected throughput: Aggregated line rate of interface(s) used in the DUT/SUT or the value defined based on requirement for a specific deployment scenario

Initial inspected throughput: 10% of "Target inspected throughput" (an optional parameter for documentation)

Number of HTTP response object requests (transactions) per connection: 10

RECOMMENDED HTTP response object size: 1, 16, 64, 256 KByte, and mixed objects defined in the table



Object size (KByte)	Number of requests/ Weight
0.2	1
6	1
8	1
9	1
10	1
25	1
26	1
35	1
59	1
347	1

Table 4: Mixed Objects

#### **7.3.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.
- b. Traffic should be forwarded constantly.
- c. Concurrent TCP connections MUST be constant during steady state and any deviation of concurrent TCP connections SHOULD be less than 10%. This confirms the DUT opens and closes TCP connections almost at the same rate.



#### **7.3.3.4. Measurement**

Inspected Throughput and HTTP Transactions per Second MUST be reported for each object size.

#### **7.3.4. Test Procedures and Expected Results**

The test procedure is designed to measure HTTP throughput of the DUT/SUT. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution and HTTP response object sizes.

##### **7.3.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "Initial inspected throughput" as defined in the parameters [Section 7.3.3.2](#).

The traffic load profile SHOULD be defined as described in [Section 4.3.4](#). The DUT/SUT SHOULD reach the "Initial inspected throughput" during the sustain phase. Measure all KPI as defined in [Section 7.3.3.4](#).

The measured KPIs during the sustain phase MUST meet the test results validation criteria "a" defined in [Section 7.3.3.3](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.3.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish the target objective ("Target inspected throughput") defined in the parameters table. The test equipment SHOULD start to measure and record all specified KPIs and the frequency of measurements SHOULD be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT is expected to reach the desired value of the target objective in the sustain phase. Follow step 3, if the measured value does not meet the target value or does not fulfill the test results validation criteria.



#### **7.3.4.3. Step 3: Test Iteration**

Determine the achievable inspected throughput within the test results validation criteria and measure the KPI metric Transactions per Second. Final test iteration MUST be performed for the test duration defined in [Section 4.3.4](#).

### **7.4. HTTP Transaction Latency**

#### **7.4.1. Objective**

Using HTTP traffic, determine the HTTP transaction latency when DUT is running with sustainable HTTP transactions per second supported by the DUT/SUT under different HTTP response object sizes.

Test iterations MUST be performed with different HTTP response object sizes in two different scenarios. One with a single transaction and the other with multiple transactions within a single TCP connection. For consistency both the single and multiple transaction test MUST be configured with HTTP 1.1.

Scenario 1: The client MUST negotiate HTTP 1.1 and close the connection with FIN immediately after completion of a single transaction (GET and RESPONSE).

Scenario 2: The client MUST negotiate HTTP 1.1 and close the connection FIN immediately after completion of 10 transactions (GET and RESPONSE) within a single TCP connection.

#### **7.4.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

#### **7.4.3. Test Parameters**

In this section, benchmarking test specific parameters SHOULD be defined.

##### **7.4.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented.



#### **7.4.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target objective for scenario 1: 50% of the connection per second measured in benchmarking test TCP/HTTP Connections Per Second ([Section 7.2](#))

Target objective for scenario 2: 50% of the inspected throughput measured in benchmarking test HTTP Throughput ([Section 7.3](#))

Initial objective for scenario 1: 10% of Target objective for scenario 1" (an optional parameter for documentation)

Initial objective for scenario 2: 10% of "Target objective for scenario 2" (an optional parameter for documentation)

HTTP transaction per TCP connection: test scenario 1 with single transaction and the second scenario with 10 transactions

HTTP 1.1 with GET command requesting a single object. The RECOMMENDED object sizes are 1, 16, and 64 KByte. For each test iteration, client MUST request a single HTTP response object size.

#### **7.4.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile. Ramp up and ramp down phase SHOULD NOT be considered.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections.



- c. During the sustain phase, traffic should be forwarded at a constant rate.
- d. Concurrent TCP connections MUST be constant during steady state and any deviation of concurrent TCP connections SHOULD be less than 10%. This confirms the DUT opens and closes TCP connections almost at the same rate.
- e. After ramp up the DUT MUST achieve the "Target objective" defined in the parameter [Section 7.4.3.2](#) and remain in that state for the entire test duration (sustain phase).

#### **[7.4.3.4](#). Measurement**

TTFB (minimum, average and maximum) and TTLB (minimum, average and maximum) MUST be reported for each object size.

#### **[7.4.4](#). Test Procedures and Expected Results**

The test procedure is designed to measure TTFB or TTLB when the DUT/SUT is operating close to 50% of its maximum achievable connections per second or inspected throughput. This test procedure MAY be repeated multiple times with different IP types (IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution), HTTP response object sizes and single and multiple transactions per connection scenarios.

##### **[7.4.4.1](#). Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "Initial objective" as defined in the parameters [Section 7.4.3.2](#). The traffic load profile can be defined as described in [Section 4.3.4](#).

The DUT/SUT SHOULD reach the "Initial objective" before the sustain phase. The measured KPIs during the sustain phase MUST meet the test results validation criteria a, b, c, d, e and f defined in [Section 7.4.3.3](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure MUST NOT be continued to "Step 2".



#### **7.4.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish "Target objective" defined in the parameters table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#).

The test equipment SHOULD start to measure and record all specified KPIs and the frequency of measurement SHOULD be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT MUST reach the desired value of the target objective in the sustain phase.

Measure the minimum, average and maximum values of TFB and TTLB.

### **7.5. Concurrent TCP/HTTP Connection Capacity**

#### **7.5.1. Objective**

Determine the number of concurrent TCP connections that the DUT/ SUT sustains when using HTTP traffic.

#### **7.5.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.

#### **7.5.3. Test Parameters**

In this section, benchmarking test specific parameters SHOULD be defined.

##### **7.5.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented.

##### **7.5.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be noted for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)



Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target concurrent connection: Initial value from product datasheet or the value defined based on requirement for a specific deployment scenario.

Initial concurrent connection: 10% of "Target concurrent connection" (an optional parameter for documentation)

Maximum connections per second during ramp up phase: 50% of maximum connections per second measured in benchmarking test TCP/HTTP Connections per second ([Section 7.2](#))

Ramp up time (in traffic load profile for "Target concurrent connection"): "Target concurrent connection" / "Maximum connections per second during ramp up phase"

Ramp up time (in traffic load profile for "Initial concurrent connection"): "Initial concurrent connection" / "Maximum connections per second during ramp up phase"

The client MUST negotiate HTTP 1.1 with persistence and each client MAY open multiple concurrent TCP connections per server endpoint IP.

Each client sends 10 GET commands requesting 1 KByte HTTP response object in the same TCP connection (10 transactions/TCP connection) and the delay (think time) between each transaction MUST be X seconds.

$$X = ("Ramp\ up\ time" + "steady\ state\ time") / 10$$

The established connections SHOULD remain open until the ramp down phase of the test. During the ramp down phase, all connections SHOULD be successfully closed with FIN.

#### **[7.5.3.3](#). Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transaction) of total attempted transactions.



- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections.
- c. During the sustain phase, traffic SHOULD be forwarded constantly.

#### **7.5.3.4. Measurement**

Average Concurrent TCP Connections MUST be reported for this benchmarking test.

#### **7.5.4. Test Procedures and Expected Results**

The test procedure is designed to measure the concurrent TCP connection capacity of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.

##### **7.5.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure test equipment to establish "Initial concurrent TCP connections" defined in [Section 7.5.3.2](#). Except ramp up time, the traffic load profile SHOULD be defined as described in [Section 4.3.4](#).

During the sustain phase, the DUT/SUT SHOULD reach the "Initial concurrent TCP connections". The measured KPIs during the sustain phase MUST meet the test results validation criteria "a" and "b" defined in [Section 7.5.3.3](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.5.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish the target objective ("Target concurrent TCP connections"). The test equipment SHOULD follow the traffic load profile definition (except ramp up time) as described in [Section 4.3.4](#).

During the ramp up and sustain phase, the other KPIs such as inspected throughput, TCP connections per second and application transactions per second MUST NOT reach to the maximum value that the DUT/SUT can support.



The test equipment SHOULD start to measure and record KPIs defined in [Section 7.5.3.4](#). The frequency of measurement SHOULD be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT is expected to reach the desired value of the target objective in the sustain phase. Follow step 3, if the measured value does not meet the target value or does not fulfill the test results validation criteria.

#### **[7.5.4.3](#). Step 3: Test Iteration**

Determine the achievable concurrent TCP connections capacity within the test results validation criteria.

### **[7.6](#). TCP/HTTPS Connections per Second**

#### **[7.6.1](#). Objective**

Using HTTPS traffic, determine the sustainable SSL/TLS session establishment rate supported by the DUT/SUT under different throughput load conditions.

Test iterations MUST include common cipher suites and key strengths as well as forward looking stronger keys. Specific test iterations MUST include ciphers and keys defined in [Section 7.6.3.2](#).

For each cipher suite and key strengths, test iterations MUST use a single HTTPS response object size defined in the test equipment configuration parameters [Section 7.6.3.2](#) to measure connections per second performance under a variety of DUT Security inspection load conditions.

#### **[7.6.2](#). Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

#### **[7.6.3](#). Test Parameters**

In this section, benchmarking test specific parameters SHOULD be defined.



#### **7.6.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented.

#### **7.6.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target connections per second: Initial value from product datasheet or the value defined based on requirement for a specific deployment scenario.

Initial connections per second: 10% of "Target connections per second" (an optional parameter for documentation)

RECOMMENDED ciphers and keys defined in [Section 4.3.1.3](#)

The client MUST negotiate HTTPS 1.1 and close the connection with FIN immediately after completion of one transaction. In each test iteration, client MUST send GET command requesting a fixed HTTPS response object size. The RECOMMENDED object sizes are 1, 2, 4, 16, and 64 KByte.

#### **7.6.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria:

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections.



- c. During the sustain phase, traffic should be forwarded at a constant rate.
- d. Concurrent TCP connections MUST be constant during steady state and any deviation of concurrent TCP connections SHOULD be less than 10%. This confirms the DUT opens and closes TCP connections almost at the same rate.

#### **7.6.3.4. Measurement**

TCP Connections Per Second MUST be reported for each test iteration (for each object size).

The KPI metric TLS Handshake Rate can be measured in the test using 1KByte object size.

#### **7.6.4. Test Procedures and Expected Results**

The test procedure is designed to measure the TCP connections per second rate of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.

##### **7.6.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "Initial connections per second" as defined in [Section 7.6.3.2](#). The traffic load profile MAY be defined as described in [Section 4.3.4](#).

The DUT/SUT SHOULD reach the "Initial connections per second" before the sustain phase. The measured KPIs during the sustain phase MUST meet the test results validation criteria a, b, c, and d defined in [Section 7.6.3.3](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.6.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish "Target connections per second" defined in the parameters table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#).



During the ramp up and sustain phase, other KPIs such as inspected throughput, concurrent TCP connections and application transactions per second MUST NOT reach the maximum value that the DUT/SUT can support. The test results for specific test iteration SHOULD NOT be reported, if the above mentioned KPI (especially inspected throughput) reaches the maximum value. (Example: If the test iteration with 64 KByte of HTTPS response object size reached the maximum inspected throughput limitation of the DUT, the test iteration can be interrupted and the result for 64 KByte SHOULD NOT be reported).

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT is expected to reach the desired value of the target objective ("Target connections per second") in the sustain phase. Follow step 3, if the measured value does not meet the target value or does not fulfill the test results validation criteria.

#### **7.6.4.3. Step 3: Test Iteration**

Determine the achievable connections per second within the test results validation criteria.

### **7.7. HTTPS Throughput**

#### **7.7.1. Objective**

Determine the sustainable inspected throughput of the DUT/SUT for HTTPS transactions varying the HTTPS response object size.

Test iterations MUST include common cipher suites and key strengths as well as forward looking stronger keys. Specific test iterations MUST include the ciphers and keys defined in the parameter [Section 7.7.3.2](#).

#### **7.7.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.



### **7.7.3. Test Parameters**

In this section, benchmarking test specific parameters SHOULD be defined.

#### **7.7.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented.

#### **7.7.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target inspected throughput: Aggregated line rate of interface(s) used in the DUT/SUT or the value defined based on requirement for a specific deployment scenario.

Initial inspected throughput: 10% of "Target inspected throughput" (an optional parameter for documentation)

Number of HTTPS response object requests (transactions) per connection: 10

RECOMMENDED ciphers and keys defined in [Section 4.3.1.3](#)

RECOMMENDED HTTPS response object size: 1, 16, 64, 256 KByte, and mixed objects defined in the table below.



Object size (KByte)	Number of requests/ Weight
0.2	1
6	1
8	1
9	1
10	1
25	1
26	1
35	1
59	1
347	1

Table 5: Mixed Objects

#### **7.7.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.
- b. Traffic should be forwarded constantly.
- c. Concurrent TCP connections MUST be constant during steady state and any deviation of concurrent TCP connections SHOULD be less than 10%. This confirms the DUT opens and closes TCP connections almost at the same rate.



#### **7.7.3.4. Measurement**

Inspected Throughput and HTTP Transactions per Second MUST be reported for each object size.

#### **7.7.4. Test Procedures and Expected Results**

The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution and HTTPS response object sizes.

##### **7.7.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "initial inspected throughput" as defined in the parameters [Section 7.7.3.2](#).

The traffic load profile should be defined as described in [Section 4.3.4](#). The DUT/SUT SHOULD reach the "Initial inspected throughput" during the sustain phase. Measure all KPI as defined in [Section 7.7.3.4](#).

The measured KPIs during the sustain phase MUST meet the test results validation criteria "a" defined in [Section 7.7.3.3](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.7.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish the target objective ("Target inspected throughput") defined in the parameters table. The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT is expected to reach the desired value of the target objective in the sustain phase. Follow step 3, if the measured value does not meet the target value or does not fulfill the test results validation criteria.



#### **7.7.4.3. Step 3: Test Iteration**

Determine the achievable average inspected throughput within the test results validation criteria. Final test iteration MUST be performed for the test duration defined in [Section 4.3.4](#).

### **7.8. HTTPS Transaction Latency**

#### **7.8.1. Objective**

Using HTTPS traffic, determine the HTTPS transaction latency when DUT is running with sustainable HTTPS transactions per second supported by the DUT/SUT under different HTTPS response object size.

Scenario 1: The client MUST negotiate HTTPS and close the connection with FIN immediately after completion of a single transaction (GET and RESPONSE).

Scenario 2: The client MUST negotiate HTTPS and close the connection with FIN immediately after completion of 10 transactions (GET and RESPONSE) within a single TCP connection.

#### **7.8.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

#### **7.8.3. Test Parameters**

In this section, benchmarking test specific parameters SHOULD be defined.

##### **7.8.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented.

##### **7.8.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)



Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

RECOMMENDED cipher suites and key sizes defined in [Section 4.3.1.3](#)

Target objective for scenario 1: 50% of the connections per second measured in benchmarking test TCP/HTTPS Connections per second ([Section 7.6](#))

Target objective for scenario 2: 50% of the inspected throughput measured in benchmarking test HTTPS Throughput ([Section 7.7](#))

Initial objective for scenario 1: 10% of Target objective for scenario 1" (an optional parameter for documentation)

Initial objective for scenario 2: 10% of "Target objective for scenario 2" (an optional parameter for documentation)

HTTPS transaction per TCP connection: test scenario 1 with single transaction and the second scenario with 10 transactions

HTTPS 1.1 with GET command requesting a single object. The RECOMMENDED object sizes are 1, 16, and 64 KByte. For each test iteration, client MUST request a single HTTPS response object size.

#### **[7.8.3.3](#). Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile. Ramp up and ramp down phase SHOULD NOT be considered.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections
- c. During the sustain phase, traffic should be forwarded at a constant rate.
- d. Concurrent TCP connections MUST be constant during steady state and any deviation of concurrent TCP connections SHOULD be less than 10%. This confirms the DUT opens and closes TCP connections almost at the same rate



- e. After ramp up the DUT MUST achieve the "Target objective" defined in the parameter [Section 7.8.3.2](#) and remain in that state for the entire test duration (sustain phase).

#### **[7.8.3.4.](#) Measurement**

TTFB (minimum, average and maximum) and TTLB (minimum, average and maximum) MUST be reported for each object size.

#### **[7.8.4.](#) Test Procedures and Expected Results**

The test procedure is designed to measure TTFB or TTLB when the DUT/SUT is operating close to 50% of its maximum achievable connections per second or inspected throughput. This test procedure MAY be repeated multiple times with different IP types (IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution), HTTPS response object sizes and single and multiple transactions per connection scenarios.

##### **[7.8.4.1.](#) Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "Initial objective" as defined in the parameters [Section 7.8.3.2](#). The traffic load profile can be defined as described in [Section 4.3.4](#).

The DUT/SUT SHOULD reach the "Initial objective" before the sustain phase. The measured KPIs during the sustain phase MUST meet the test results validation criteria a, b, c, d, e and f defined in [Section 7.8.3.3](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **[7.8.4.2.](#) Step 2: Test Run with Target Objective**

Configure test equipment to establish "Target objective" defined in the parameters table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#).

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT MUST reach the desired value of the target objective in the sustain phase.



Measure the minimum, average and maximum values of TFB and TTLB.

## **[7.9.](#) Concurrent TCP/HTTPS Connection Capacity**

### **[7.9.1.](#) Objective**

Determine the number of concurrent TCP connections that the DUT/SUT sustains when using HTTPS traffic.

### **[7.9.2.](#) Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

### **[7.9.3.](#) Test Parameters**

In this section, benchmarking test specific parameters SHOULD be defined.

#### **[7.9.3.1.](#) DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific benchmarking test MUST be documented.

#### **[7.9.3.2.](#) Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this benchmarking test:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

RECOMMENDED cipher suites and key sizes defined in [Section 4.3.1.3](#)

Target concurrent connections: Initial value from product datasheet or the value defined based on requirement for a specific deployment scenario.

Initial concurrent connections: 10% of "Target concurrent connections" (an optional parameter for documentation)



Connections per second during ramp up phase: 50% of maximum  
connections per second measured in benchmarking test TCP/HTTPS  
Connections per second ([Section 7.6](#))

Ramp up time (in traffic load profile for "Target concurrent  
connections"): "Target concurrent connections" / "Maximum  
connections per second during ramp up phase"

Ramp up time (in traffic load profile for "Initial concurrent  
connections"): "Initial concurrent connections" / "Maximum  
connections per second during ramp up phase"

The client MUST perform HTTPS transaction with persistence and each  
client can open multiple concurrent TCP connections per server  
endpoint IP.

Each client sends 10 GET commands requesting 1 KByte HTTPS response  
objects in the same TCP connections (10 transactions/TCP connection)  
and the delay (think time) between each transaction MUST be X  
seconds.

$$X = ("Ramp\ up\ time" + "steady\ state\ time") / 10$$

The established connections SHOULD remain open until the ramp down  
phase of the test. During the ramp down phase, all connections  
SHOULD be successfully closed with FIN.

#### **[7.9.3.3.](#) Test Results Validation Criteria**

The following test Criteria is defined as test results validation  
criteria. Test results validation criteria MUST be monitored during  
the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP  
response code other than 200 OK) MUST be less than 0.001% (1 out  
of 100,000 transactions) of total attempted transactions.
- b. Number of Terminated TCP connections due to unexpected TCP RST  
sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000  
connections) of total initiated TCP connections.
- c. During the sustain phase, traffic SHOULD be forwarded constantly.

#### **[7.9.3.4.](#) Measurement**

Average Concurrent TCP Connections MUST be reported for this  
benchmarking test.



#### **7.9.4. Test Procedures and Expected Results**

The test procedure is designed to measure the concurrent TCP connection capacity of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.

##### **7.9.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure test equipment to establish "initial concurrent TCP connections" defined in [Section 7.9.3.2](#). Except ramp up time, the traffic load profile SHOULD be defined as described in [Section 4.3.4](#).

During the sustain phase, the DUT/SUT SHOULD reach the "Initial concurrent TCP connections". The measured KPIs during the sustain phase MUST meet the test results validation criteria "a" and "b" defined in [Section 7.9.3.3](#).

If the KPI metrics do not meet the test results validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.9.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish the target objective ("Target concurrent TCP connections"). The test equipment SHOULD follow the traffic load profile definition (except ramp up time) as described in [Section 4.3.4](#).

During the ramp up and sustain phase, the other KPIs such as inspected throughput, TCP connections per second and application transactions per second MUST NOT reach to the maximum value that the DUT/SUT can support.

The test equipment SHOULD start to measure and record KPIs defined in [Section 7.9.3.4](#). The frequency of measurement SHOULD be less than 2 seconds. Continue the test until all traffic profile phases are completed.

Within the test results validation criteria, the DUT/SUT is expected to reach the desired value of the target objective in the sustain phase. Follow step 3, if the measured value does not meet the target value or does not fulfill the test results validation criteria.



#### **7.9.4.3. Step 3: Test Iteration**

Determine the achievable concurrent TCP connections within the test results validation criteria.

### **8. IANA Considerations**

The IANA has allocated 2001:0200::/48 for IPv6 testing, which is a 48-bit prefix from the [\[RFC4733\]](#) pool. For IPv4 testing, the IP subnet 198.18.0.0/15 has been assigned to the BMWG by the IANA. This assignment was made to minimize the chance of conflict in case a testing device were to be accidentally connected to part of the Internet. The specific use of the IPv4 addresses is detailed in [\[RFC2544\] Appendix C](#).

### **9. Security Considerations**

The primary goal of this document is to provide benchmarking terminology and methodology for next-generation network security devices. However, readers should be aware that there is some overlap between performance and security issues. Specifically, the optimal configuration for network security device performance may not be the most secure, and vice-versa. The Cipher suites recommended in this document are just for test purpose only. The Cipher suite recommendation for a real deployment is outside the scope of this document.

### **10. Contributors**

The following individuals contributed significantly to the creation of this document:

Alex Samonte, Amritam Putatunda, Aria Eslambolchizadeh, David DeSanto, Jurrie Van Den Breekel, Ryan Liles, Samaresh Nair, Stephen Goudreault, and Tim Otto

### **11. Acknowledgements**

The authors wish to acknowledge the members of NetSecOPEN for their participation in the creation of this document. Additionally, the following members need to be acknowledged:

Anand Vijayan, Baski Mohan, Chao Guo, Chris Brown, Chris Marshall, Jay Lindenauer, Michael Shannon, Mike Deichman, Ray Vinson, Ryan Riese, Tim Carlin, and Toulmay Orkun



## **12. References**

### **12.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

### **12.2. Informative References**

- [RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", [RFC 2544](#), DOI 10.17487/RFC2544, March 1999, <<https://www.rfc-editor.org/info/rfc2544>>.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), DOI 10.17487/RFC2616, June 1999, <<https://www.rfc-editor.org/info/rfc2616>>.
- [RFC2647] Newman, D., "Benchmarking Terminology for Firewall Performance", [RFC 2647](#), DOI 10.17487/RFC2647, August 1999, <<https://www.rfc-editor.org/info/rfc2647>>.
- [RFC3511] Hickman, B., Newman, D., Tadjudin, S., and T. Martin, "Benchmarking Methodology for Firewall Performance", [RFC 3511](#), DOI 10.17487/RFC3511, April 2003, <<https://www.rfc-editor.org/info/rfc3511>>.
- [RFC4733] Schulzrinne, H. and T. Taylor, "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals", [RFC 4733](#), DOI 10.17487/RFC4733, December 2006, <<https://www.rfc-editor.org/info/rfc4733>>.
- [RFC6815] Bradner, S., Dubray, K., McQuaid, J., and A. Morton, "Applicability Statement for [RFC 2544](#): Use on Production Networks Considered Harmful", [RFC 6815](#), DOI 10.17487/RFC6815, November 2012, <<https://www.rfc-editor.org/info/rfc6815>>.



[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [RFC 8446](#), DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

## **Appendix A. Test Methodology - Security Effectiveness Evaluation**

### **A.1. Test Objective**

This test methodology verifies the DUT/SUT is able to detect, prevent and report the vulnerabilities.

In this test, background test traffic will be generated in order to utilize the DUT/SUT. In parallel, the CVEs will be sent to the DUT/SUT as encrypted and as well as clear text payload formats using a traffic generator. The selection of the CVEs is described in [Section 4.2.1](#).

- o Number of blocked CVEs
- o Number of bypassed (nonblocked) CVEs
- o Background traffic performance (verify if the background traffic is impacted while sending CVE toward DUT/SUT)
- o Accuracy of DUT/SUT statistics in term of vulnerabilities reporting

### **A.2. Test Bed Setup**

The same Test bed MUST be used for security effectiveness test and as well as for benchmarking test cases defined in [Section 7](#).

### **A.3. Test Parameters**

In this section, the benchmarking test specific parameters SHOULD be defined.

#### **A.3.1. DUT/SUT Configuration Parameters**

DUT/SUT configuration Parameters MUST conform to the requirements defined in [Section 4.2](#). The same DUT configuration MUST be used for Security effectiveness test and as well as for benchmarking test cases defined in [Section 7](#). The DUT/SUT MUST be configured in inline mode and all detected attack traffic MUST be dropped and the session Should be reset



#### **A.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). The same Client and server IP ranges MUST be configured as used in the benchmarking test cases. In addition, the following parameters MUST be documented for this benchmarking test:

- o Background Traffic: 45% of maximum HTTP throughput and 45% of Maximum HTTPS throughput supported by the DUT/SUT (measured with object size 64 KByte in the benchmarking tests "HTTP(S) Throughput" defined in [Section 7.3](#) and [Section 7.7](#).
- o RECOMMENDED CVE traffic transmission Rate: 10 CVEs per second
- o RECOMMEND to generate each CVE multiple times (sequentially) at 10 CVEs per second
- o Ciphers and Keys for the encrypted CVE traffic MUST use the same cipher configured for HTTPS traffic related benchmarking tests ([Section 7.6](#) - [Section 7.9](#))

#### **A.4. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole test duration.

- a. Number of failed Application transaction in the background traffic MUST be less than 0.01% of attempted transactions
- b. Number of Terminated TCP connections of the background traffic (due to unexpected TCP RST sent by DUT/SUT) MUST be less than 0.01% of total initiated TCP connections in the background traffic
- c. During the sustain phase, traffic should be forwarded at a constant rate
- d. False positive MUST NOT occur in the background traffic

#### **A.5. Measurement**

Following KPI metrics MUST be reported for this test scenario:

Mandatory KPIs:

- o Blocked CVEs: It should be represented in the following ways:



- \* Number of blocked CVEs out of total CVEs
- \* Percentage of blocked CVEs
- o Unblocked CVEs: It should be represented in the following ways:
  - \* Number of unblocked CVEs out of total CVEs
  - \* Percentage of unblocked CVEs
- o Background traffic behavior: it should represent one of the followings ways:
  - \* No impact (traffic transmission at a constant rate)
  - \* Minor impact (e.g. small spikes- +/- 100 Mbit/s)
  - \* Heavily impacted (e.g. large spikes and reduced the background HTTP(S) throughput > 100 Mbit/s)
- o DUT/SUT reporting accuracy: DUT/SUT MUST report all detected vulnerabilities.

Optional KPIs:

- o List of unblocked CVEs

#### **[A.6.](#) Test Procedures and Expected Results**

The test procedure is designed to measure the security effectiveness of the DUT/SUT at the sustaining period of the traffic load profile. The test procedure consists of two major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.

##### **[A.6.1.](#) Step 1: Background Traffic**

Generate the background traffic at the transmission rate defined in the parameter section.

The DUT/SUT MUST reach the target objective (HTTP(S) throughput) in sustain phase. The measured KPIs during the sustain phase MUST meet the test results validation criteria a, b, c and d defined in [Appendix A.4](#).

If the KPI metrics do not meet the acceptance criteria, the test procedure MUST NOT be continued to "Step 2".



### **A.6.2. Step 2: CVE Emulation**

While generating the background traffic (in sustain phase), send the CVE traffic as defined in the parameter section.

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement MUST be less than 2 seconds. Continue the test until all CVEs are sent.

The measured KPIs MUST meet all the test results validation criteria a, b, c, and d defined in [Appendix A.4](#).

In addition, the DUT/SUT SHOULD report the vulnerabilities correctly.

### **Appendix B. DUT/SUT Classification**

This document attempts to classify the DUT/SUT in four different four different categories based on its maximum supported firewall throughput performance number defined in the vendor datasheet. This classification MAY help user to determine specific configuration scale (e.g., number of ACL entries), traffic profiles, and attack traffic profiles, scaling those proportionally to DUT/SUT sizing category.

The four different categories are Extra Small, Small, Medium, and Large. The RECOMMENDED throughput values for the following categories are:

Extra Small (XS) - supported throughput less than 1Gbit/s

Small (S) - supported throughput less than 5Gbit/s

Medium (M) - supported throughput greater than 5Gbit/s and less than 10Gbit/s

Large (L) - supported throughput greater than 10Gbit/s

#### **Authors' Addresses**

Balamuhunthan Balarajah  
Berlin  
Germany

Email: [bm.balarajah@gmail.com](mailto:bm.balarajah@gmail.com)



Carsten Rossenhoevel  
EANTC AG  
Salzufer 14  
Berlin 10587  
Germany

Email: [cross@eantc.de](mailto:cross@eantc.de)

Brian Monkman  
NetSecOPEN  
417 Independence Court  
Mechanicsburg, PA 17050  
USA

Email: [bmonkman@netsecopen.org](mailto:bmonkman@netsecopen.org)

