Benchmarking Methodology Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

B. Balarajah
C. Rossenhoevel
EANTC AG

March 5, 2018

# Benchmarking Methodology for Network Security Device Performance
# draft-balarajah-bmwg-ngfw-performance-02

## Abstract

This document provides benchmarking terminology and methodology for next-generation network security devices including next-generation firewalls (NGFW), intrusion detection and prevention solutions (IDS/IPS) and unified threat management (UTM) implementations. The 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 application use cases. The main areas covered in this document are test terminology, traffic profiles and benchmarking methodology for NGFWs to start with.

## Status of This Memo

## Copyright Notice

Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . 3
2. Requirements . . . . . . . . . . . . . . . . . . . . . . . . 3
3. Scope . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
4. Test Setup . . . . . . . . . . . . . . . . . . . . . . . . . 3
  4.1. Testbed Configuration . . . . . . . . . . . . . . . . . . 4
  4.2. DUT/SUT Configuration . . . . . . . . . . . . . . . . . . 5
  4.3. Test Equipment Configuration . . . . . . . . . . . . . . . 8
    4.3.1. Client Configuration . . . . . . . . . . . . . . . . . 9
    4.3.2. Backend Server Configuration . . . . . . . . . . . . . 10
    4.3.3. Traffic Flow Definition . . . . . . . . . . . . . . . 11
    4.3.4. Traffic Load Profile . . . . . . . . . . . . . . . . . 12
5. Test Bed Considerations . . . . . . . . . . . . . . . . . . . 13
6. Reporting . . . . . . . . . . . . . . . . . . . . . . . . . . 14
  6.1. Key Performance Indicators . . . . . . . . . . . . . . . . 15
7. Benchmarking Tests . . . . . . . . . . . . . . . . . . . . . 16
  7.1. Throughput Performance With NetSecOPEN Traffic Mix . . . 16
    7.1.1. Objective . . . . . . . . . . . . . . . . . . . . . . 16
    7.1.2. Test Setup . . . . . . . . . . . . . . . . . . . . . . 17
    7.1.3. Test Parameters . . . . . . . . . . . . . . . . . . . 17
    7.1.4. Test Procedures and expected Results . . . . . . . . 19
  7.2. Concurrent TCP Connection Capacity With HTTP Traffic . . 20
    7.2.1. Objective . . . . . . . . . . . . . . . . . . . . . . 20
    7.2.2. Test Setup . . . . . . . . . . . . . . . . . . . . . . 20
    7.2.3. Test Parameters . . . . . . . . . . . . . . . . . . . 20
    7.2.4. Test Procedures and expected Results . . . . . . . . 22
  7.3. TCP/HTTP Connections Per Second . . . . . . . . . . . . . 23
    7.3.1. Objective . . . . . . . . . . . . . . . . . . . . . . 23
  7.4. HTTP Transactions Per Second . . . . . . . . . . . . . . 24
    7.4.1. Objective . . . . . . . . . . . . . . . . . . . . . . 24
  7.5. HTTP Throughput . . . . . . . . . . . . . . . . . . . . . 24
    7.5.1. Objective . . . . . . . . . . . . . . . . . . . . . . 24
  7.6. HTTP Transaction Latency . . . . . . . . . . . . . . . . 24
    7.6.1. Objective . . . . . . . . . . . . . . . . . . . . . . 24
  7.7. Concurrent SSL/TLS Connection Capacity . . . . . . . . . 24
    7.7.1. Objective . . . . . . . . . . . . . . . . . . . . . . 24
  7.8. SSL/TLS Handshake Rate . . . . . . . . . . . . . . . . . 24
    7.8.1. Objective . . . . . . . . . . . . . . . . . . . . . . 24
  7.9. HTTPS Transaction  Per Second . . . . . . . . . . . . . . 25
    7.9.1. Objective . . . . . . . . . . . . . . . . . . . . . . 25
  7.10. HTTPS Throughput . . . . . . . . . . . . . . . . . . . . 25
    7.10.1. Objective . . . . . . . . . . . . . . . . . . . . . 25

7.11. HTTPS Transaction Latency . . . . . . . . . . . . . . . . . 25
7.11.1. Objective . . . . . . . . . . . . . . . . . . . . . . 25
8. Formal Syntax . . . . . . . . . . . . . . . . . . . . . . . 25
9. IANA Considerations . . . . . . . . . . . . . . . . . . . . 25
10. Security Considerations . . . . . . . . . . . . . . . . . . 25
11. Acknowledgements . . . . . . . . . . . . . . . . . . . . . 26
12. Normative References . . . . . . . . . . . . . . . . . . . 26
Appendix A. An Appendix . . . . . . . . . . . . . . . . . . . 26
Authors' Addresses . . . . . . . . . . . . . . . . . . . . . . 26

## 1. Introduction

15 years have passed since IETF recommended test methodology and terminology for firewalls initially (RFC 2647, RFC 3511). 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: They enable fair and reasonable comparison of network security functions. All these reasons have led to the creation of a new next-generation firewall benchmarking document.

## 2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] .

## 3. Scope

This document provides testing terminology and testing methodology next-generation firewalls and related security functions. It covers two main areas: Performance benchmarks and security effectiveness testing. The 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 will be applicable to all of the benchmarking test cases described in Section 7.

### 4.1. Testbed Configuration

Testbed 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 number of physical links and forwarding performance capabilities (throughput and latency) of the network devise in the testbed. For this reason, this document recommends to avoid external devices such as switch and router in the testbed as possible.

In the typical deployment, the security devices (DUT/SUT) will not have a large number of entries in MAC or ARP tables, which impact the actual DUT/SUT performance due to MAC and ARP table lookup processes. Therefore, depend on number of used IP address in client and server side, it is recommended to connect Layer 3 device(s) between test equipment and DUT/SUT as shown in Figure 1.

If the test equipment is capable to emulate layer 3 routing functionality and there is no need for test equipment ports aggregation, it is recommended to configure the test setup as shown in Figure 2.

```
    +-------------------+      +-----------+      +--------------------+
    |Aggregation Switch/|      |           |      | Aggregation Switch/|
    | Router            +------+  DUT/SUT  +------+ Router             |
    |                   |      |           |      |                    |
    +----------+--------+      +-----------+      +--------+-----------+
               |                                           |
               |                                           |
   +-----------+-----------+                   +-----------+-----------+
   |                       |                   |                       |
   | +-------------------+ |                   | +-------------------+ |
   | | Emulated Router(s)| |                   | | Emulated Router(s)| |
   | |     (Optional)    | |                   | |     (Optional)    | |
   | +-------------------+ |                   | +-------------------+ |
   | +-------------------+ |                   | +-------------------+ |
   | |      Clients      | |                   | |      Servers      | |
   | +-------------------+ |                   | +-------------------+ |
   |                       |                   |                       |
   |    Test Equipment     |                   |    Test Equipment     |
   +-----------------------+                   +-----------------------+
```

Figure 1: Testbed Setup - Option 1

```
   +-----------------------+                   +-----------------------+
   | +-------------------+ |   +-----------+   | +-------------------+ |
   | | Emulated Router(s)| |   |           |   | | Emulated Router(s)| |
   | |    (Optional)     | +----- DUT/SUT  +-----+    (Optional)     | |
   | +-------------------+ |   |           |   | +-------------------+ |
   | +-------------------+ |   +-----------+   | +-------------------+ |
   | |     Clients       | |                   | |      Servers      | |
   | +-------------------+ |                   | +-------------------+ |
   |                       |                   |                       |
   |   Test Equipment      |                   |   Test Equipment      |
   +-----------------------+                   +-----------------------+
```

Figure 2: Testbed Setup - Option 2

## 4.2. DUT/SUT Configuration

An unique DUT/SUT configuration MUST be used for all of the benchmarking tests described in Section 7. Since each DUT/SUT will have their own unique configuration, users SHOULD configure their device with the same parameters that would be used in the actual deployment of the device or a typical deployment. Also it is mandatory to enable all the security features on the DUT/SUT in order to achieve maximum security coverage for a specific deployment scenario.

This document attempts to define the recommended security features which SHOULD be consistently enabled for all test cases. The table below describes the recommended sets of feature list which SHOULD be configured on the DUT/SUT. In order to improve repeatability, a summary of the DUT configuration including description of all enabled DUT/SUT features MUST be published with the benchmarking results.

| | Device | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | NGFW | | | NGIPS | AD | WAF | BPS | SSL Broker |
| DUT Features | Feature | Included in initial Scope | Added to future Scope | Future test standards to be developed | | | | |
| SSL Inspection | x | | x | | | | | |
| IDS/IPS | x | x | | | | | | |
| Web Filtering | x | | x | | | | | |
| Antivirus | x | x | | | | | | |
| Anti Spyware | x | x | | | | | | |
| Anti Botnet | x | x | | | | | | |
| DLP | x | | x | | | | | |
| DDoS | x | | x | | | | | |
| Certificate Validation | x | | x | | | | | |
| Logging and Reporting | x | x | | | | | | |
| Application Identification | x | x | | | | | | |

Table 1: DUT/SUT Feature List

In addition, it is also recommended to configure a realistic number of access policy rules on the DUT/SUT. This document determines the number of access policy rules for three different class of DUT/SUT. The classification of the DUT/SUT MAY be based on its maximum supported throughput performance number. This document classifies the DUT/SUT in three different categories; namely small, medium and maximum.

The recommended throughput values for the following classes are;

Small - supported throughput less than 5Gbit/s

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

Large - supported throughput greater than 10Gbit/s

The access rule defined in the table 2 MUST be configured from top to bottom in correct order.  The configured access policy rule MUST NOT block the test traffic used for the benchmarking test scenario.

| | | | | DUT/SUT Classification # Rules | | |
|---|---|---|---|---|---|---|
| Rules Type | Match Criteria | Description | Action | Small | Medium | Large |
| Application layer | Application | Any application traffic NOT included in the test traffic | block | 10 | 20 | 50 |
| Transport layer | Src IP and TCP/UDP Dst ports | Any src IP use in the test AND any dst ports NOT used in the test traffic | block | 50 | 100 | 250 |
| IP layer | Src/Dst IP | Any src/dst IP NOT used in the test | block | 50 | 100 | 250 |
| Application layer | Application | Applications included in the test traffic | allow | 10 | 10 | 10 |
| Transport layer | Src IP and TCP/UDP Dst ports | Half of the src IP used in the test AND any dst ports used in the test traffic. One rule per subnet | allow | 1 | 1 | 1 |
| IP layer | Src IP | The rest of the src IP subnet range used in the test. One rule per subnet | allow | 1 | 1 | 1 |

Table 2: DUT/SUT Access List

### 4.3. Test Equipment Configuration

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

This document attempts to explicitly specify which test equipment parameters SHOULD be configurable, any such parameter(s) MUST be noted in the test report.

### 4.3.1. Client Configuration

This section specifies which parameters SHOULD be considerable while configuring emulated 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 TCP Reno variant, which include congestion avoidance, back off and windowing, retransmission and recovery on every TCP connection between client and server endpoints. The default IPv4 and IPv6 MSS segments size MUST be set to 1460 bytes and 1440 bytes and a TX and RX receive windows of 32768 bytes. Delayed ACKs are permitted, but it SHOULD be limited to either a 200 msec delay timeout or 3000 in bytes before a forced ACK. Up to 3 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.

#### 4.3.1.2. Client IP Address Space

The sum of the client IP space SHOULD contain the following attributes. The traffic blocks SHOULD consist of multiple unique, continuous static address blocks. A default gateway is permitted. The IPv4 ToS byte should be set to '00'.

The following equation can be used to determine the required total number of client IP address.

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

(Idea 1) 6-7 Mbps per IP= 1,400-1,700 IPs per 10Gbit/s throughput

(Idea 2) 0.1-0.2 Mbps per IP = 50,000-100,000 IPs per 10Gbit/s throughput

Based on deployment and usecase scenario, client IP addresses SHOULD be distributed between IPv4 and IPv6 type. This document recommends using the following ratio(s) between IPv4 and IPv6:

(Idea 1) 100 % IPv4, no IPv6

(Idea 2) 80 % IPv4, 20 % IPv6

(Idea 3) 50 % IPv4, 50 % IPv6

(Idea 4) 0 % IPv4, 100 % IPv6

#### 4.3.1.3. Emulated Web Browser Attributes

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

For HTTP traffic emulation, the emulated browser must negotiate HTTP 1.1. HTTP persistency MAY be enabled depend on test scenario. The browser CAN 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 can be processed if the emulated browser has available connections. The browser MUST advertise a User-Agent header. Headers will be sent uncompressed. The browser should enforce content length validation.

For encrypted traffic, the following attributes shall define the negotiated encryption parameters. The tests must use TLSv1.2 or higher with a record size of 16383, commonly used cipher suite and key strength. Session reuse or ticket resumption may be used for subsequent connections to the same Server endpoint IP. The client endpoint must send TLS Extension SNI information when opening up a security tunnel. Server certificate validation should be disabled. Server certificate validation should be disabled. Cipher suite and certificate size should be defined in the parameter session of benchmarking tests.

### 4.3.2. Backend Server Configuration

This document attempts to specify which parameters should be considerable while configuring emulated backend servers using test equipment.

#### 4.3.2.1. TCP Stack Attributes

The TCP stack SHOULD use a TCP Reno variant, which include congestion avoidance, back off and windowing, retransmission and recovery on every TCP connection between client and server endpoints. The default IPv4 MSS segment size MUST be set to 1460 bytes and a TX and RX receive windows of at least 32768 bytes. Delayed ACKs are permitted but SHOULD be limited to either a 200 msec delay timeout or 3000 in bytes before a forced ACK. Up to 3 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.

#### 4.3.2.2. Server Endpoint IP Addressing

The server IP blocks should consist of unique, continuous static address blocks with one IP per Server FQDN endpoint per test port. The IPv4 ToS byte should be set to '00'. The source mac address of the server endpoints shall be the same emulating routed behavior. Each Server FQDN should have it's own unique IP address. The Server IP addressing should be fixed to the same number of FQDN entries.

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

The emulated server pool for HTTP should listen on TCP port 80 and emulated HTTP version 1.1 with persistence. For HTTPS server, the pool must have the same basic attributes of an HTTP server pool plus attributes for SSL/TLS. The server must advertise a server type. For HTTPS server, TLS 1.2 or higher must be used with a record size of 16383 bytes and ticket resumption or Session ID reuse enabled. The server must listen on port TCP 443. The server shall serve a certificate to the client. It is required that the HTTPS server also check Host SNI information with the Fully Qualified Domain Name (FQDN). Client certificate validation should be disabled. Cipher suite and certificate size should be defined in the parameter session of benchmarking tests.

### 4.3.3. Traffic Flow Definition

The section describes the traffic pattern between the client and server endpoints. At the beginning of the test, the server endpoint initializes and will be in a ready to accept connection state 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 the MAC and IP address. The behavior of the client is to sweep though 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 port server 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 shall use Fully Qualified Domain Names in Host Headers and for TLS 1.2 Server Name Indication (SNI).

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

Client endpoints are independent of other clients that are concurrently executing. When a client endpoint initiate traffic, this section will describe how the steps though different services. Once initialized, the user should randomly hold (perform no operation) for a few milliseconds to allow for better randomization of start of client traffic. The client will then either open up a new TCP connection or connect to a TCP persistence stack still open to that specific server. At any point that the service profile may require encryption, a TLS 1.2 encryption tunnel will form presenting the URL request to the server. The server will 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 object. The initial object to the server may not have a fixed size; its size is based on benchmarking tests described in Section 7. Multiple additional sub-URLs (Objects on the service page) may be requested simultaneously. This may or may not 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 will be described in this section. The loading of an traffic load profile has five distinct phases: Init, ramp up, sustain, ramp down/close, and collection.

Within 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 resolution shall the test iteration move to the next phase. No measurements are made in this phase. The minimum recommended 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.

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 do not overwhelm DUT/SUT's supported performance metrics namely; connection setup rate, concurrent connection and application transaction. The recommended time duration for the ramp up phase is 180- 300 seconds. No measurements are made in this phase.

In the sustain phase, the test equipment should keep to generate traffic t constant target value for a constant number of active client IPs.  The recommended time duration for sustain phase is 600 seconds.  This is the phase where measurements occur.

In the ramp down/close phase, no new connection is established and no measurements are made.  The recommend duration of this phase is between 180 to 300 seconds.

The last phase is administrative and will be when the tester 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 whole test session.  A number of factors might influence stability specifically for virtualized test beds, for example additional work loads 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 desired traffic generator aspects such as maximum throughput and the network performance metrics such as maximum latency and maximum packet loss are met.

Once the desired maximum performance goals for the system under test have been identified, a safety margin of 10 % SHOULD be added for throughput and subtracted for maximum latency and maximum packet loss.

Test bed preparation may be performed either by configuring the DUT in the most trivial setup (fast forwarding) or without presence of 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 result sections. The following attributes should be present in the introduction section of the test report.

1. The name of the NetSecOPEN traffic mix must be prominent.

2. The time and date of the execution of the test must be prominent.

3. Summary of testbed software and Hardware details

   A. DUT Hardware/Virtual Configuration

      + This section should clearly identify the make and model of the DUT

      + iThe port interfaces, including speed and link information must be documented.

      + If the DUT is a virtual VNF, interface acceleration such as DPDK and SR-IOV must be documented as well as cores used, RAM used, and the pinning / resource sharing configuration. The Hypervisor and version must be documented.

      + Any additional hardware relevant to the DUT such as controllers must be documented

   B. DUT Software

      + The operating system name must be documented

      + The version must be documented

      + The specific configuration must be documented

   C. DUT Enabled Features

      + Specific features, such as logging, NGFW, DPI must be documented

      + iAttributes 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

4. Results Summary / Executive Summary

   1. 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.

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

      a. KPIs must be documented separately for each test scenario. The format of the KPI metrics should be presented as described in Section 6.1.

      b. The next level of detains 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.

## 6.1. Key Performance Indicators

This section lists KPIs for overall benchmarking tests scenarios. All KPIs MUST be measured in whole period of sustain phase as described in Section 4.3.4. All KPIs MUST be measured from test equipment's result output.

o TCP Concurrent Connection
This key performance indicator will measure the average concurrent open TCP connections in the sustaining period.

o TCP Connection Setup Rate
This key performance indicator will measure the average established TCP connections per second in the sustaining period. For Session setup rate benchmarking test scenario, the KPI will measure average established and terminated TCP connections per second simultaneously.

o Application Transaction Rate
This key performance indicator will measure the average successful transactions per seconds in the sustaining period.

- TLS Handshake Rate
  This key performance indicator will measure the average TLS 1.2 or higher session formation rate within the sustaining period.

- Throughput
  This key performance indicator will measure the average Layer 1 throughput within the sustaining period as well as average packets per seconds within the same period. The value of throughput should be presented in Gbps rounded to two places of precision with a more specific kbps in parenthesis. Optionally, goodput may also be logged as an average goodput rate measured over the same period. Goodput result shall also be presented in the same format as throughput.

- URL Response time / Time to Last Byte (TTLB)
  This key performance indicator will measure the minimum, average and maximum per URL response time in the sustaining period as well as the average variance in the same period.

- Application Transaction Time
  This key performance indicator will measure the minimum, average and maximum the amount of time to receive all objects from the server.

- Time to First Byte (TTFB)
  This key performance indicator will measure minimum, average and maximum the time to first byte. TTFB is the elapsed time between sending the SYN packet from the client and receiving the first byte of application date from the DUT/SUT. TTFB SHOULD be expressed in millisecond.

- TCP Connect Time
  This key performance indicator will measure minimum, average and maximum TCP connect time. It is elapsed between the time the client sends a SYN packet and the time it receives the SYN/ACK. TCP connect time SHOULD be expressed in millisecond.

## 7. Benchmarking Tests

### 7.1. Throughput Performance With NetSecOPEN Traffic Mix

#### 7.1.1. Objective

To determine the average throughput performance of the DUT/SUT when using application traffic mix defined in Section 7.1.3.3.

### 7.1.2. Test Setup

Test bed setup MUST be configured as defined in Section 4. Any test scenario specific test bed configuration changes must be documented.

### 7.1.3. Test Parameters

In this section, test scenario 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 test scenario MUST be documented.

#### 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 test scenario:

Client IP address range

Server IP address range

Traffic distribution ratio between IPv4 and IPv6

Traffic load objective or specification type (e.g. Throughput, SimUsers and etc.)

Target throughput: It MAY be defined based on requirements. Otherwise it represents aggregated line rate of interface(s) used in the DUT/SUT

Initial throughput: Initial throughput MAY be up to 10% of the "Target throughput"

#### 7.1.3.3. Traffic Profile

Test scenario MUST be run with a single application traffic mix profile. The name of the NetSecOpen traffic mix MUST be documented.

#### 7.1.3.4. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance criteria

a. Number of failed Application transaction MUST be 0.01%.

b. Number of Terminated TCP connection due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.01%

c. Maximum deviation (max. dev) of application transaction time / TTLB (Time To Last Byte) MUST be less than X (The value for "X" will be finalyzed and updated in future draft release)
   The following equation MUST be used to calculate the deviation of application transaction time or TTLB.

   max. dev = max((avg_latency - min_latency),(max_latency - avg_latency)) / (Initial latency)

   Where, the initial latency is calculated using the following equation. For this calculation, the latency values (min', avg' and max') MUST be measured during test procedure step 1 as defined in Section 7.1.4.1.
   The variable latency represents application transaction time or TTLB.

   Initial latency:= min((avg' latency - min' latency) | (max' latency - avg' latency))

d. Maximum value of TCP connect time must be less than Xms (The value for "X" will be finalyzed and updated in future draft release). The definition for TCP connect time is found in Section 6.1.

e. Maximum value of Time to First Byte must be less than 2* TCP connect time.

Test Acceptance criteria for this test scenario MUST be monitored during the sustain phase of the traffic load profile only.

### 7.1.3.5. Measurement

Following KPI metrics MUST be reported for this test scenario.

Mandatory KPIs: average Throughput, maximum Concurrent TCP connection, TTLB/application transaction time (minimum, average and maximum) and average application transaction rate

Optional KPIs: average TCP connection setup rate, average TLS handshake rate, TCP connect time and TTFB

### 7.1.4. Test Procedures and expected Results

The test procedure is designed to measure the throughput performance of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps.

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

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

Configure traffic load profile of the test equipment to generate test traffic at "initial throughput" rate as described in the parameters section. The DUT/SUT SHOULD reach the "initial 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 acceptance criteria "a" and "b" defined in Section 7.1.3.4.

If the KPI metrics do not meet the acceptance 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 "Target 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. The frequency of KPI metrics measurement MUST be less than 5 seconds. Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target throughput during the sustain phase. In addition, the measured KPIs must meet all acceptance criteria. Follow the step 3, if the KPI metrics do not meet the acceptance criteria.

#### 7.1.4.3. Step 3: Test Iteration with Binary Search

Use binary search algorithm to configure the desired traffic load profile for each test iteration. Binary search algorithmn can be implemented using the parameter; Resolution =0.01* Target throughput and Backoff= 50%.

Determine the maximum and average achievable throughput within the acceptance criteria.

### 7.2. Concurrent TCP Connection Capacity With HTTP Traffic

#### 7.2.1. Objective

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

#### 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, test scenario 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 test scenario 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 noted for this test scenario:

Client IP address range

Server IP address range

Traffic distribution ratio between IPv4 and IPv6

Traffic load objective or specification type (e.g Throughput, SimUsers and etc.)

Target concurrent connection: It can be defined based on requirements

Initial concurrent connection: 10% of "Target concurrent connection"

#### 7.2.3.2.1. Client Configuration Parameters

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

Test scenario SHOULD be run with a single traffic profile with following attributes:

HTTP 1.1 with GET command requesting 10 Kbyte objects with random MIME type.

The test equipment SHOULD perform HTTP transactions within each TCP connection subsequently.  The frequency of transactions MUST be defined to achieve X% of total throughput that DUT can support.  The suggested value of X is 25.  It will be finalyzed and updated in the next draft version.

During the sustain state of concurrent connection and traffic load , a minimal % of TCP connection SHOULD be closed and re-opened.

### 7.2.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance criteria

a.  Number of failed Application transaction MUST be less than 0.01% of attempt transaction.

b.  Number of Terminated TCP connection due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.01% of total initiated TCP sessions.

c.  During the sustain phase, traffic should be forwarded constantly at the rate defined in the parameter Section 7.2.3.

d.  Maximum deviation (max. dev) of application transaction time / TTLB (Time To Last Byte) MUST be less than Xms (The value for "X" will be finalyzed and updated in future draft release).
    The following equation MUST be used to calculate the deviation of application transaction time or TTLB.

    max. dev = max((avg_latency - min_latency),(max_latency - avg_latency)) / (Initial latency)

    Where, the initial latency is calculated using the following equation.  For this calculation, the latency values (min', avg' and max') MUST be measured during test procedure step 1 as defined in Section 7.1.4.1.

The variable latency represents application transaction time or TTLB.

Initial latency:= min((avg' latency - min' latency) | (max' latency - avg' latency))

e. Maximum value of TCP connect time must be less than Xms (The value for "X" will be finalyzed and updated in future draft release). The definition for TCP connect time is found in Section 6.1.

f. Maximum value of Time to First Byte must be less than 2* TCP connect time.

Test Acceptance criteria for this test scenario MUST be monitored during the sustain phase of the traffic load profile only.

#### 7.2.3.4. Measurement

Following KPI metrics MUST be reported for this test scenario;

average Throughput, max. Min. Avg. Concurrent TCP connection, TTLB/ application transaction time (minimum, average and maximum) and average application transaction rate.

### 7.2.4. 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.

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

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

Configure traffic load profile of the test equipment to establish "initial concurrent connection" as defined in the parameters section. The traffic load profile should be defined as described in Section 4.3.4.

The DUT/SUT SHOULD reach the "initial concurrent connection" during the sustain phase. The measured KPIs during the sustain phase MUST meet the acceptance criteria "a" and "b" defined in Section 7.2.3.3

If the KPI metrics do not meet the acceptance 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 "Target concurrent connection" 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, the other KPIs such as throughput, TCP connection rate and application transaction MUST NOT reach to the maximum value that the DUT/SUT can support. Throughput, TCP connection rate and application transaction should not be reached more than X% of maximum value that DUT can support. The suggested value of X is 25. It will be finalyzed and updated in the next draft version.

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement MUST be less than 5 seconds. Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target concurrent connection at the sustain phase. In addition, the measured KPIs must meet all acceptance criteria.

Follow the step 3, if the KPI metrics do not meet the acceptance criteria.

#### 7.2.4.3. Step 3: Test Iteration with Binary Search

Use binary search algorithm to configure the desired traffic load profile for each test iteration. Binary search algorithmn can be implemented using the parameter; Resolution =0.01* "Target concurrent connection" and Backoff= 50%.

Determine the maximum and average achievable throughput within the acceptance criteria.

### 7.3. TCP/HTTP Connections Per Second

#### 7.3.1. Objective

Using HTTP traffic, determine the maximum and average value of TCP session establishment rate supported by the DUT/SUT.

Test parameters and test test procedures will be added in the future release.

### 7.4. HTTP Transactions Per Second

#### 7.4.1. Objective

Determine maximum and average HTTP transacton rate supported by the DUT/SUT.

Test parameters and test test procedures will be added in the future release.

### 7.5. HTTP Throughput

#### 7.5.1. Objective

Determine the average throughput performance of the DUT/SUT when using HTTP traffic.

Test parameters and test test procedures will be added in the future release.

### 7.6. HTTP Transaction Latency

#### 7.6.1. Objective

Determine the minimum, average and maximum values of HTTP transaction latency at 80% throughput rate measured in "HTTP Throughput" test scenario.

Test parameters and test test procedures will be added in the future release.

### 7.7. Concurrent SSL/TLS Connection Capacity

#### 7.7.1. Objective

Usin encrypted traffic (HTTPS), determine the maximum number of concurrent TCP connection that DUT/SUT sustains.

Test parameters and test test procedures will be added in the future release.

### 7.8. SSL/TLS Handshake Rate

#### 7.8.1. Objective

Determine the maximum and average SSL/TLS handshake rate supported by the DUT/SUT.

Test parameters and test test procedures will be added in the future release.

### 7.9. HTTPS Transaction Per Second

#### 7.9.1. Objective

Determine maximum and average HTTPS transacton rate supported by the DUT/SUT.

Test parameters and test test procedures will be added in the future release.

### 7.10. HTTPS Throughput

#### 7.10.1. Objective

Determine the average throughput performance of the DUT/SUT when using HTTPS traffic.

Test parameters and test test procedures will be added in the future release.

### 7.11. HTTPS Transaction Latency

#### 7.11.1. Objective

Determine the minimum, average and maximum values of HTTPS transaction latency at 80% throughput rate measured in "HTTPS Throughput" test scenario.

Test parameters and test test procedures will be added in the future release.

## 8. Formal Syntax

## 9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

## 10. Security Considerations

Security consideration will be added in the future release.

## 11. Acknowledgements

Acknowledgements will be added in the future release.

## 12. 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>.

## Appendix A. An Appendix

Details about NetSecOPEN traffic mix will be added in next draft release.

Authors' Addresses

Balamuhunthan Balarajah
EANTC AG
Salzufer 14
Berlin  10587
Germany

Email: balarajah@eantc.de

Carsten Rossenhoevel
EANTC AG
Salzufer 14
Berlin  10587
Germany

Email: cross@eantc.de