TEST METHODOLOGY

Data Center Network Security (DCNS)

October 10, 2018

V2.0

Table of Contents

1 Introduction

1.1 Data Center Network Security

Enterprises demand a lot of their data centers, which makes their performance and availability paramount. Infrastructure and application architectures are designed to work in concert with each other, and if any component is incorrectly sized or configured, this has the potential to disrupt or impact applications for employees or customers. Network security technology is essential in a data center architecture, providing connectivity and in some cases traffic inspection or special handling to protect critical assets in the data center.

Data center network security (DCNS) is a term used to describe a class of devices that provide network security for the data center. There are several device types in this category; the data center firewall (DCFW) and the data center intrusion prevention system (DCIPS) are the most well-known, each having been deployed for a number of years. A third type of device combines the capabilities of the DCFW and DCIPS and is referred to as a data center security gateway (DCSG).

When considering a data center network security device, performance metrics become critical. The volume of traffic will be significantly higher than for a device that is intended to protect end users within the corporate network perimeter. Data center network security devices handle traffic for potentially hundreds of thousands of users who are accessing large applications in a server farm. Application traffic generates many connections and transactions per request, which places a high demand on a network security device's ability to set-up many connections quickly, hold many connections open, and achieve high throughput rates.

1.2 The Need for the Data Center Firewall

Firewall technology is one of the largest and most mature security markets. Firewalls have undergone several stages of development, from early packet filtering and circuit relay firewalls to application layer (proxy-based) and dynamic packet filtering firewalls. Throughout their history, however, the goal has been to enforce an access control policy between two networks, and thus firewalls should be viewed as an implementation of policy. Firewalls are mechanisms used to protect trusted networks from untrusted networks while allowing authorized communications to pass from one side to the other.

1.3 The Need for the Data Center Intrusion Prevention System

The data center intrusion prevention system is an important component of data center network security. Designed to identify and block attacks against web servers, application servers, and database servers, a DCIPS can provide temporary protection and relief from the immediate need to patch affected systems. The DCIPS must catch sophisticated attacks while producing nearly zero false positives, and it must not significantly degrade network performance or it will never be installed.

1.4 The Need for the Data Center Security Gateway

Data center security gateways are the convergence of data center security capabilities, and as such, provide a vital role in today's security infrastructure. The DCSG must be capable of performing deep packet inspection on all

packets and ports and over all protocols in order to determine which applications are running over which ports and thus secure them effectively. High availability, low latency, and fault tolerance are mission-critical.

1.5 About This Test Methodology

NSS Labs' test reports are designed to address the challenges faced by enterprise security and IT professionals in selecting and managing security products. The scope of this particular methodology includes:

- Security effectiveness
- Performance
- Stability and reliability
- Total cost of ownership (TCO)

Since DCNS devices are deployed at critical choke points in the network, their stability and reliability is imperative. Therefore, regardless of any features or capabilities, the main requirement of any DCNS device is that it must be as stable, reliable, fast, and flexible as the network it protects.

NSS Labs test methodologies are continually evolving in response to feedback. If you would like to provide input, please contact advisor@nsslabs.com. For a list of changes, please reference the Change Log in the Appendix.

1.6 NSS Labs Group Test Inclusion Criteria

In order to encourage the greatest participation, and to allay any potential concerns of bias, NSS invites all security vendors claiming DCNS capabilities to submit their products at no cost. Vendors with major market share, as well as challengers with new technology, will be included.

The DCNS device should be supplied as a single appliance where possible (cluster controller solutions are also acceptable), with the appropriate number of physical interfaces capable of achieving the vendor-specified level of connectivity and throughput. NSS' minimum test requirement for a DCNS device is one inline 10 Gbps port pair per 10 Gbps of unidirectional throughput (20 Gbps UDP bidirectional throughput). The maximum number of ports will be connected to determine the overall throughput of the DCNS device. Both SFP+ and QSFP+ transceivers are acceptable (or a combination of both types of interfaces). For devices with QSFP+ transceivers, QSFP+ breakout cables will utilized. Management interfaces must be 1 Gbps copper.

	SFP+ Transceivers (1 x 10 Gbps)	**QSFP+ Transceivers (4 x 10 Gbps)**
Minimum Interfaces	2 x 10 Gbps: 10 Gbps unidirectional throughput 20 Gbps UDP bidirectional throughput	1 x 40 Gbps: 20 Gbps unidirectional throughput 40 Gbps UDP bidirectional throughput
Maximum Interfaces	96 x 10 Gbps: 480 Gbps unidirectional throughput 960 Gbps UDP bidirectional throughput	24 x 40 Gbps: 480 Gbps unidirectional throughput 960 Gbps UDP bidirectional throughput

Figure 1 – NSS' Minimum and Maximum Testing Requirements for a DCNS Device

Once installed in the test lab, the DCNS device will be configured for the use case appropriate to the target deployments (corporate network data center). DCFW and DCSG devices must be configured to block all traffic when resources are exhausted or when traffic cannot be analyzed for any reason (often referred as "failing

closed"). A DCIPS device is expected to allow all traffic through when a failure occurs (commonly called "failing open"). The DCIPS device should be configurable and will have bypass (fail-open) capabilities verified during the power failure test. After this, it is expected to fail closed or drop traffic when resources are exhausted to enable identification of performance limits. A DCIPS that allows traffic when maximums are exceeded may miss attacks, and this will be reflected in its scoring.

DCFW and DCSG devices will be implemented as layer three (routing) devices. A DCIPS will be implemented as a transparent inline device.

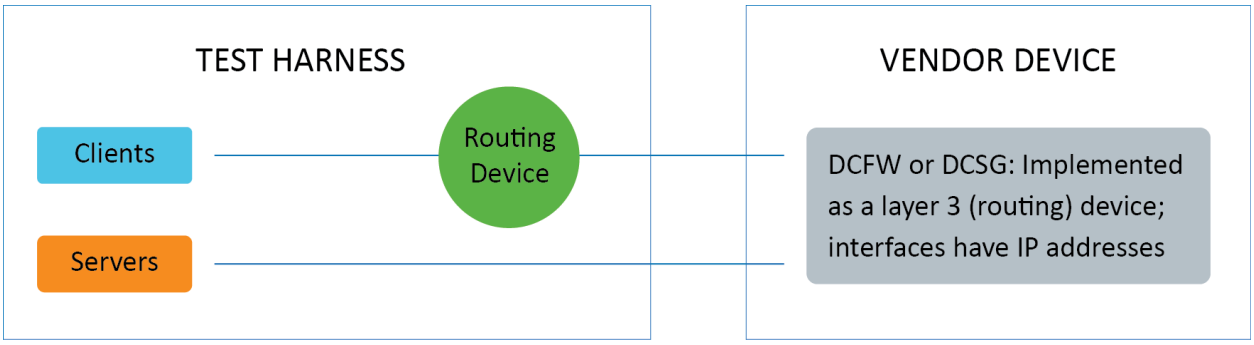

Figure 2 – DCFW or DCSG Configuration

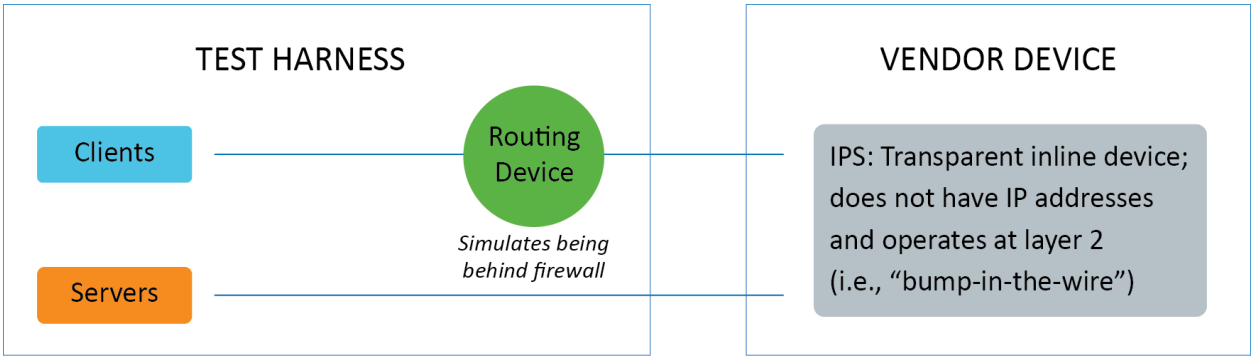

Figure 3 – DCIPS Configuration

2 Product Guidance

NSS issues summary product guidance based on evaluation criteria that is important to information security professionals. The evaluation criteria are as follows:

- Security effectiveness
- Resistance to evasion
- Stability and reliability
- Performance
- Total Cost of Ownership (TCO)

Products are listed in rank order according to their guidance rating.

2.1 Recommended

A *Recommended* rating from NSS indicates that a product has performed well and deserves strong consideration. Only the top technical products earn a *Recommended* rating from NSS, regardless of market share, company size, or brand recognition.

2.2 Neutral

A *Neutral* rating from NSS indicates that a product has performed reasonably well and should continue to be used if it is the incumbent within an organization. Products that earn a *Neutral* rating from NSS deserve consideration during the purchasing process.

2.3 Caution

A *Caution* rating from NSS indicates that a product has performed poorly. Organizations using one of these products should review their security posture and other threat mitigation factors, including possible alternative configurations and replacement. Products that earn a *Caution* rating from NSS should not be short-listed or renewed.

3 Security Effectiveness

This section describes the security effectiveness tests in which the DCNS device will be evaluated. Once testing begins, the product version and configuration will be frozen to preserve the integrity of the test. A dedicated management interface (virtual or otherwise) is preferred.

Note that security effectiveness will be tested over IPv4.

3.1 Firewall Policy Enforcement (DCFW and DCSG only)

The DCFW must support stateful firewalling either by managing state tables to prevent "traffic leakage" or as a stateful proxy. The DCFW must be able to manage firewall policy across multiple interfaces/zones. NSS also requires that a single security policy be applied to all interfaces under test. At a minimum, the firewall must provide a "trusted" internal interface and an "untrusted" external/Internet interface.

Policies are rules that are configured on a firewall to permit or deny access from one network resource to another, based on identifying criteria such as source, destination, and service. Policies are typically written to permit or deny network traffic from one or more of the following zones:

- **Untrusted –** This is typically an external network and is considered to be unknown and not secure. An example of an untrusted network would be the Internet.
- **Trusted –** This is typically an internal network, which is a network that is considered secure and protected.

NSS' firewall tests verify performance and the ability to enforce policy between the following:

- Trusted to Untrusted
- Untrusted to Trusted

3.1.1 Baseline Policy

Routed configuration with an "allow all" policy

3.1.2 Simple Policies

Simple outbound and inbound policies allow basic browsing and email access for internal clients and no external access

3.1.3 Complex Policies

Complex outbound and inbound policies consist of many rules, objects, and services.

3.1.4 Static NAT (Network Address Translation)

Inbound NAT using fixed IP address translation with one-to-one mapping

3.1.5 SYN Flood Protection

The basis of a SYN flood attack is to fail to complete the three-way handshake necessary to establish a legitimate session. The objective of SYN flooding is to disable one side of the TCP connection, which will result in one or more of the following:

- The server is unable to accept new connections.
- The server crashes or becomes inoperative.
- Authorization between servers is impaired.

The device is expected to protect against SYN floods.

3.1.6 IP Address Spoofing

This test attempts to confuse the firewall into allowing traffic to pass from one network segment to another. By forging the IP header to contain a source address that is different from the address that transmitted the packet, an attacker can make it appear that the packet was sent from a different (trusted) machine. The endpoint that receives successfully spoofed packets will respond to the forged source address (the attacker).

The device is expected to protect against IP address spoofing.

3.2 Intrusion Prevention (DCIPS and DCSG only)

These are policies consisting of threat protection signatures that verify whether the IPS component is capable of correctly blocking malicious traffic based on a comparison of packet/session contents against signatures/filters/protocol decoders.

Performance and security effectiveness are measured using a tuned policy. Once testing begins, the product version and configuration will be frozen to preserve the integrity of the test. Vendors are encouraged to assist NSS engineers in tuning the device to optimize protection and eliminate false positives.

All signatures used must be available to the general public at the time of testing; no custom signatures are permitted. NSS has found that this approach reflects a typical enterprise deployment and will align results from testing with product performance in the field. The device is required to block and log exploit attempts and hostile traffic.

Performance and security effectiveness are measured using a tuned policy. Vendors are permitted and encouraged to assist NSS engineers in tuning the device to eliminate false positives and provide an appropriate level of protection for the target environment.

Tuning: Security engineers typically will tune an IPS to ensure its protection coverage matches the needs of the environment in which it is placed.

3.2.1 False Positive Testing

The ability of the device to identify and allow legitimate traffic while maintaining protection against threats and exploits is just as important as its abilility to protect against malicious content. This test will include a varied sample of legitimate nonmalicious application traffic that should be allowed.

3.2.2 Exploit Library

NSS' security effectiveness testing leverages the deep expertise of our engineers who utilize multiple commercial, open-source, and proprietary tools. With thousands of exploits, this is the industry's most comprehensive test to date. Most notably, all of the live exploits and payloads in the NSS exploit test have been validated in our lab such that one or more of the following is true:

- A reverse shell is returned
- A bind shell is opened on the target allowing the attacker to execute arbitrary commands
- Arbitrary code is executed
- A malicious payload is installed
- A system is rendered unresponsive

This test goes far beyond replaying packet captures or pressing the button on a test tool. In short, NSS engineers trigger vulnerabilities to validate that an exploit was able to pass through the device.

3.2.3 Coverage by Attack Vector

Threats and exploits can be initiated by either the target or the attacker targeting local or remote vulnerabilities.

3.2.3.1 Attacker Initiated

Also referred to as "server-side" exploits, the threat/exploit is executed remotely by the attacker against a vulnerable application and/or operating system.

3.2.3.2 Network

Threats and exploits are initiated as a result of network communication.

3.2.4 Coverage by Impact Type

The *NSS Exploit Library* contains thousands of publicly available exploits (including multiple variants of each exploit), from which groups of exploits are carefully selected to test based on appropriate usage. Each exploit has been validated to impact the target vulnerable host(s).

3.2.4.1 System Exposure

Attacks resulting in remote system compromise, which provide the attacker with the ability to execute arbitrary system-level commands. Most of the exploits in this class are weaponized and offer the attacker a fully interactive remote shell on the target client or server.

3.2.4.2 Service Exposure

These attacks result in an individual service compromise, but they do not always provide the attacker with the ability to execute arbitrary, system-level commands, nor do they always immediately result in full system-level access to the operating system and all services. However, by using additional localized system attacks, it may be possible for the attacker to move from the service level to the system level.Typical attacks in this category include service-specific attacks, such as SQL injection, that enable the attacker to execute arbitrary SQL commands within the database service.

3.2.4.3 System or Service Fault

Attacks resulting in a system- or service-level fault that crashes the targeted service or application and requires administrative action to restart the service or reboot the system. These attacks do not enable the attacker to

execute arbitrary commands. However, the resulting impact to the business could be severe given that the attacker could crash the protected system or service.

3.2.5 Coverage by Date

NSS' security effectiveness testing will include exploits current at the time of the test, as well as target vulnerabilities covering multiple years dating backwards from the time of the test.

3.2.6 Coverage by Application Vendor

NSS' exploit test contains many vendors, including but not limited to:

- 3Com
- Apache
- Cisco
- EMC
- HP
- ISC
- ManageEngine
- MySQL
- Oracle
- SAP
- Sun Microsystems
- AlienVault
- BigAnt
- Citrix
- Fortinet
- IBM
- LanDesk
- Mercury
- Novell
- RealNetworks
- SolarWinds
- Symantec
- Alt-N
- CA
- Digium
- GE
- IPSwitch
- Lighttpd
- Microsoft
- Nullsoft
- Samba
- Squid

3.3 Evasions

Attackers can modify basic attacks to evade detection in a number of ways. If a device fails to detect a single form of evasion, any exploit can pass through the device, rendering it ineffective. NSS verifies that the device is capable of detecting and blocking basic exploits when subjected to varying common evasion techniques. Wherever possible, the device is expected to successfully decode the obfuscated traffic in order to provide an accurate alert relating to the original exploit, rather than alerting purely on anomalous traffic detected as a result of the evasion technique itself.

A number of common exploits are executed across the device to ensure that they are detected in their unmodified state. These will be chosen from a suite of older/common basic exploits for which NSS is certain that all vendors will have signatures. None of the exploits that were used in section 3.2 will be used as evasion baselines. This ensures that vendors are not provided with any information on the content of any part of the main *NSS Exploit Library* in advance of the test.

3.3.1 IP Packet Fragmentation

These tests determine the effectiveness of the fragment reassembly mechanism of the device.

- Fragments from 8 – 32 bytes in size
- Ordered, out-of-order, or reverse order fragments
- Fragment overlap, favoring new and favoring old data
- Interleaved, duplicate, duplicate with or without incrementing DWORD, duplicate packets with random payload, or duplicate packets scheduled for later delivery
- Any combination of the above methods

3.3.2 Stream Segmentation

These tests determine the effectiveness of the stream reassembly mechanism of the device.

- Segments from 1 – 2048 bytes in size
- Ordered, reverse ordered, or out-of-order segments, with “favor old” or “favor new”
- Duplicate, duplicate interleaved, duplicate last packet, or overlapping segments
- Invalid or NULL TCP control flags
- Sequence resync requests, random initial sequence number, or out-of-window sequence numbers
- Faked retransmits, PAWS elimination, or segments containing random data
- Endianness interchanged
- Any combination of the above methods

3.3.3 RPC Fragmentation

Both Sun/ONC RPC and MS-RPC allow the sending application to fragment requests, and all MS-RPC services have a built-in fragmentation reassembly mechanism.

An attacker can transmit the BIND followed by a single request fragmented over a hundred actual requests with small fragments of the malicious payload. Alternatively, the attacker could transmit both the BIND and request fragments in one large TCP segment, thus foiling any signatures that use a simple size check.

NSS uses test tools that combine large writes with many tiny MS-RPC fragments and provide multiple levels of fragmentation.

These tests determine the effectiveness of the RPC reassembly mechanism of the device:

- Sun/Open Network Computing (ONC) RPC byte fragmentation
- Fragments sent in one or more TCP segments, with or without last fragment
- RPC fragmentation may or may not be performed in a single segment
- Any combination of the above methods

3.3.4 URL Obfuscation

Random URL encoding techniques are employed to transform simple URLs that are often used in pattern-matching signatures into apparently meaningless strings of escape sequences and expanded path characters, using a combination of the following techniques:

- Escape encoding (% encoding)
- Microsoft %u encoding
- Path character transformations and expansions (/./ , //, \ )

These techniques are combined in various ways for each URL tested, ranging from minimal to extreme transformation (every character transformed). All transformed URLs are verified to ensure they still function as expected:

- URL encoding – Levels 1 – 8 (minimal to extreme)
- Premature URL ending
- Long URL
- Fake parameter
- TAB separation
- Case sensitivity
- Windows\delimiter
- Session splicing
- Any combination of the above methods

3.3.5 FTP/Telnet Evasion

When attempting FTP exploits, it is possible to evade some products by inserting additional spaces and Telnet control sequences in FTP commands.

These tests insert a range of valid Telnet control sequences that can be parsed and handled by IIS FTP server and wu-ftpd, and which also conform to Section 2.3 of RFC 959. Control opcodes are inserted at random, ranging from minimal insertion (only one pair of opcodes), to extreme (opcodes between every character in the FTP command):

- Inserting spaces in FTP command lines
- Inserting non-text Telnet opcodes – Levels 1 – 8 (minimal to extreme)
- Any combination of the above methods

3.3.6 Layered Evasions

This test attempts to bypass the device by performing any legitimate combination of the evasion techniques specified in section 3.3. It will be verified that the target machine's standard network stack is capable of decoding the evasion correctly while maintaining the exploit viability.

4 Performance (All DCNS Devices)

This section measures the performance of the DCNS device using various traffic conditions that provide metrics for real-world performance. Individual implementations will vary based on usage; however, these quantitative metrics provide a gauge as to whether a particular DCNS device is appropriate for a given environment.

Note that performance will be tested over IPv4 and IPv6.

4.1 Raw Packet Processing Performance (UDP Throughput)

This test uses UDP packets of varying sizes generated by traffic generation appliances. A constant stream of packets of a specified size, with varying source and destination IP addresses, transmitting from a fixed source port to a fixed destination port, is transmitted bi-directionally through each port pair of the device.

Each packet contains dummy data and is targeted at a valid port on a valid IP address on the target subnet. The percentage load and frames per second (fps) figures across each inline port pair are verified by network monitoring tools before each test begins.

No TCP sessions are created during this test, and there is very little for the detection engine to do. However, each vendor will be required to write a signature to detect the test packets to ensure that they are being passed through the detection engine and are not being "fast-pathed."

The goal of this test is to determine the raw packet processing capability of each inline port pair of the device as well as its effectiveness at forwarding packets quickly in order to provide the highest level of network performance and with the lowest latency.

4.1.1 64 Byte Packets

Maximum 1,488,000 frames per second per Gigabit of traffic. This test determines the ability of a device to process packets from the wire under the most challenging packet processing conditions.

4.1.2 128 Byte Packets

Maximum 844,000 frames per second per Gigabit of traffic

4.1.3 256 Byte Packets

Maximum 452,000 frames per second per Gigabit of traffic.

4.1.4 512 Byte Packets

Maximum 234,000 frames per second per Gigabit of traffic. This test provides a reasonable indication of the ability of a device to process packets from the wire on an "average" network.

4.1.5 1024 Byte Packets

Maximum 119,000 frames per second per Gigabit of traffic.

4.1.6 1514 Byte Packets

Maximum 81,000 frames per second per Gigabit of traffic. This test has been included to demonstrate how easy it is to achieve good results using large packets. Readers should use caution when taking into consideration those test results that quote only performance figures that use similar packet sizes.

4.2 Latency

The purpose of the latency test is to determine the amount of time it takes for network traffic to pass through a device under a range of load conditions.

4.2.1 64 Byte Frames

Maximum 1,488,000 frames per second per Gigabit of traffic

4.2.2 128 Byte Frames

Maximum 844,000 frames per second per Gigabit of traffic

4.2.3 256 Byte Packets

Maximum 452,000 frames per second per Gigabit of traffic.

4.2.4 512 Byte Packets

Maximum 234,000 frames per second per Gigabit of traffic.

4.2.5 1024 Byte Packets

Maximum 119,000 frames per second per Gigabit of traffic.

4.2.6 1514 Byte Packets

Maximum 81,000 frames per second per Gigabit of traffic.

4.3 HTTP 1.1 Capacity

The aim of these tests is to stress the HTTP detection engine and determine how the device copes with network loads of varying average packet size and varying connections per second. By creating genuine session-based traffic with varying session lengths, the device is forced to track valid TCP sessions, thus ensuring a higher workload than for simple packet-based background traffic. This provides a test environment that is as close to real-world conditions as it is possible to achieve in a lab environment, while ensuring absolute accuracy and repeatability.

Each transaction consists of a single HTTP GET request, and there are no transaction delays (i.e., the web server responds immediately to all requests). All packets contain valid payload (a mix of binary and ASCII objects) and address data, and this test provides an excellent representation of a live network (albeit one biased towards HTTP traffic) at various network loads.

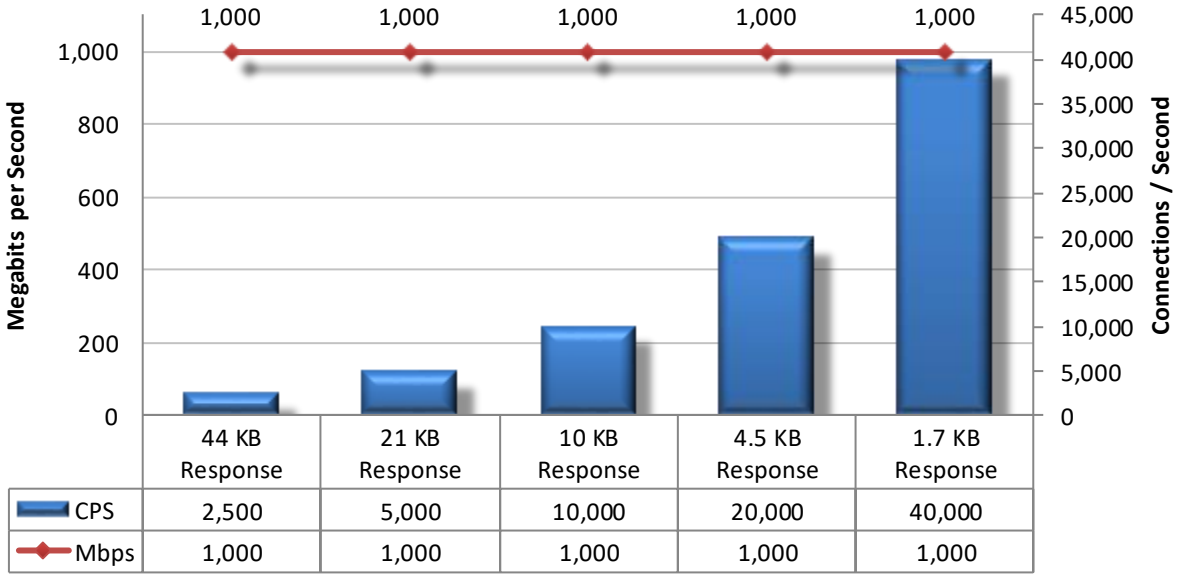

Figure 4 – HTTP Capacity

4.3.1 44 KB HTTP Response Size – 2,500 Connections per Second

Maximum 2,500 new connections per second per Gigabit of traffic with a 44 KB HTTP response size—maximum 140,000 packets per second per Gigabit of traffic. With relatively low connection rates and large packet sizes, all devices should be capable of performing well throughout this test.

4.3.2 21 KB HTTP Response Size – 5,000 Connections per Second

Maximum 5,000 new connections per second per Gigabit of traffic with a 21 KB HTTP response size—maximum 185,000 packets per second per Gigabit of traffic. With average connection rates and average packet sizes, this is a good approximation of a real-world production network, and all devices should be capable of performing well throughout this test.

4.3.3 10 KB HTTP Response Size – 10,000 Connections per Second

Maximum 10,000 new connections per second per Gigabit of traffic with a 10 KB HTTP response size—maximum 225,000 packets per second per Gigabit of traffic. With smaller packet sizes coupled with high connection rates, this represents a very heavily used production network.

4.3.4 4.5 KB HTTP Response Size – 20,000 Connections per Second

Maximum 20,000 new connections per second per Gigabit of traffic with a 4.5 KB HTTP response size—maximum 300,000 packets per second per Gigabit of traffic. With small packet sizes and extremely high connection rates, this is an extreme test for any device.

4.3.5 1.7 KB HTTP Response Size – 40,000 Connections per Second

Maximum 40,000 new connections per second per Gigabit of traffic with a 1.7 KB HTTP response size—maximum 445,000 packets per second per Gigabit of traffic. With small packet sizes and extremely high connection rates, this is an extreme test for any device.

4.4 HTTP Capacity with HTTP Persistent Connections

The aim of these tests is to determine how the device copes with network loads of varying average packet size and varying connections per second while inspecting all traffic. By creating genuine session-based traffic with varying session lengths, the device is forced to track valid TCP sessions, thus ensuring a higher workload than for simple packet-based background traffic. This provides a test environment that is as close to real-world conditions as it is possible to achieve in a lab environment, while ensuring absolute accuracy and repeatability.

This test will use HTTP persistent connections, with each TCP connection containing 10 HTTP GETs and associated responses. All packets contain valid payload (a mix of binary and ASCII objects) and address data, and this test provides an excellent representation of a live network at various network loads. The stated response size is the total of all HTTP responses within a single TCP session.

4.4.1 250 Connections per Second

This test will simulate HTTP persistent connections, each containing a total of 10 HTTP GET/responses of various sizes. The total HTTP response size for each persistent connection will be equal to four megabits, transmitted over a maximum of 250 connections per second for each gigabit of traffic.

4.4.2 500 Connections per Second

This test will simulate HTTP persistent connections, each containing a total of HTTP 10 GET/responses of various sizes. The total HTTP response size for each persistent connection will be equal to two megabits, transmitted over a maximum of 500 connections per second for each gigabit of traffic.

4.4.3 1000 Connections per Second

This test will simulate HTTP persistent connections, each containing a total of 10 HTTP GETs/responses of various sizes. The total HTTP response size for each persistent connection will be equal to one megabit, transmitted over a maximum of 1000 connections per second, for each gigabit of traffic.

4.5 Application Average Response Time: HTTP

Test traffic is passed across the infrastructure switches and through all inline port pairs of the device simultaneously (the latency of the basic infrastructure is known and is constant throughout the tests). The results are recorded for each response size (44 KB, 21 KB, 10 KB, 4.5 KB, and 1.7 KB HTTP responses) at the maximum throughput as previously determined in section 4.3 and section 4.4.

4.6 "Real-World" Single Application Flows

NSS measures a device's performance based on popular data center application profiles. Each application category has specific protocol and transaction characteristics. Within these categories, there may be one or more variants which are run to a device's failure point or up to the maximum rate of the specific flow.

- NSS_DCNS_Single_App_Video – Line Rate up to 120 Gbps
- NSS_DCNS_Single_App_HTTP_Video – Line Rate up to 120 Gbps
- NSS_DCNS_Single_App_DB – Line Rate up to 120 Gbps
- NSS_DCNS_Single_App_FIX – Line Rate up to 120 Gbps
- NSS_DCNS_Single_App_FTP – Line Rate up to 120 Gbps

- NSS_DCNS_Single_App_SMTP – Line Rate up to 120 Gbps

4.6.1 Maximum TCP Connections per Second

This test is designed to determine the maximum TCP connection rate of the device with one byte of data passing across the connections. This type of traffic would not typically be found on a normal network, but it provides the means to determine the maximum possible TCP connection rate.

The rate of establishing new sessions is increased through the device until the point when the device is not able to successfully create new connections or one of the breaking points defined earlier is reached. Each session is opened normally, one byte of data is passed to the host, and the session is closed immediately.

4.6.2 Maximum HTTP Connections per Second

This test is designed to determine the maximum TCP connection rate of the device with a one-byte HTTP response size. The response size defines the number of bytes contained in the body, excluding any bytes associated with the HTTP header. A one-byte response size is designed to provide a theoretical maximum HTTP connections per second rate.

The client and server are using HTTP 1.0 without keep-alive; the client will open a TCP connection, send one HTTP request, and then close the connection. This ensures that all TCP connections are closed immediately upon the request being satisfied; thus any concurrent TCP connections will be caused purely as a result of latency the device introduces on the network. The TCP connection rate is increased until one or more of the breaking points defined earlier is reached.

4.6.3 Maximum HTTP Transactions per Second

This test is designed to determine the maximum HTTP transaction rate of the device with a one-byte HTTP response size. The object size defines the number of bytes contained in the body, excluding any bytes associated with the HTTP header. A one-byte response size is designed to provide a theoretical maximum connections per second rate.

The client and server are using HTTP 1.1 with persistence; the client will open a TCP connection, send 10 HTTP requests, and then close the connection. This ensures that TCP connections remain open until all 10 HTTP transactions are complete, thus eliminating the maximum connection per second rate as a bottleneck (one TCP connection = 10 HTTP transactions). Load is increased until one or more of the breaking points defined earlier is reached.

4.6.4 Theoretical Maximum Concurrent TCP Connections

This test is designed to determine the maximum concurrent TCP connections of the device with no data passing across the connections. This type of traffic would not typically be found on a normal network, but it provides the means to determine the maximum possible concurrent connections figure.

An increasing number of Layer 4 TCP sessions are opened through the device. Each session is opened normally and then held open for the duration of the test as additional sessions are added up to the maximum possible. Load is increased until no more connections can be established, and this number is recorded.

4.6.5 Theoretical Maximum Concurrent TCP Connections with Data

This test is identical to section 4.6.4, except that once a connection has been established, data is transmitted (in 1 KB segments). This ensures that the device is capable of passing data across the connections once they have been established.

5 Stability and Reliability

Long-term stability is particularly important for an inline device where failure can produce network outages. These tests verify the stability of the device along with its ability to maintain security effectiveness while under normal load and while attempting to pass prohibited traffic.

The device is required to remain operational and stable throughout these tests, sustain legitimate traffic, and block 100% of prohibited traffic.

If any prohibited traffic is allowed through the device, caused by either the volume of traffic or by the device failing open for any reason, this will be recorded as a FAIL.

Note that stability and reliability will be tested over IPv4 and IPv6.

5.1 Passing Legitimate Traffic Under Extended Load with Attacks

A continuous stream of security policy violations and legitimate traffic is transmitted through the device at a rate not to exceed 75% of the device's measured capacity for eight hours. The device is expected to remain operational and to drop or block all security policy violations. This test measures the consistency of the device's blocking performance while allowing legitimate traffic in a scenario where its resources are highly utilized. The device will be considered to have failed if any of the following occurs:

- Any security policy violations are circumvented
- Legitimate traffic is dropped
- The device fails open

For devices with an IPS, security violations will include previously blocked exploits. NSS expects each exploit to be blocked, alerted on, and logged.

5.2 Behavior of the State Engine Under Load (DCSG and DCIPS only)

This test determines whether the device is capable of preserving state across a large number of open connections over an extended time period.

At various points throughout the test (including after the maximum has been reached), it is confirmed that the device is still capable of inspecting and blocking traffic that is in violation of the currently applied security policy, whilst confirming that legitimate traffic is not blocked (perhaps as a result of exhaustion of the resources allocated to state tables). The device must be able to apply policy decisions effectively based on inspected traffic at all load levels.

5.2.1 State Preservation – Normal Load

This test determines if the device maintains the state of pre-existing sessions as the number of open sessions reaches 75% of the maximum determined in section 4.6.4. A legitimate HTTP session is opened and the first packet of a two-packet exploit is transmitted. As the number of open connections approaches the maximum, the initial HTTP session is completed with the second half of the exploit and the session is closed. If the sensor is still maintaining state on the original session, the exploit should be blocked and recorded. If the state tables have been exhausted, the exploit string will be seen as a non-stateful attack and will thus be ignored. Both halves of the

exploit are required to trigger an alert. A product will fail the test if it fails to generate an alert after the second packet is transmitted, or if it raises an alert on either half of the exploit on its own.

5.2.2 State Preservation – Maximum Exceeded

This test determines whether the device maintains the state of pre-existing sessions as the number of open sessions exceeds the maximum determined in section 4.6.4. The method of execution is identical to section 5.2.1

6 Leakage Testing

This test will run TCP traffic up to 70% of the device's maximum traffic capacity and then run an attack repeatedly for eight hours. Any miss during this period constitutes a failure.

6.1 Protocol Fuzzing and Mutation

This test stresses the protocol stacks of the device by exposing it to traffic from various protocol randomizer and mutation tools. Several of the tools in this category are based on the ISIC suite and other well-known test tools. Traffic load is a maximum of 350 Mbps and 60,000 packets per second. Results are presented as PASS/FAIL. The device is expected to remain operational and capable of detecting and blocking exploits throughout the test.

6.2 Power Fail

6.2.1 DCFW and DCSG

Power to the device is cut while passing a mixture of legitimate and disallowed traffic. A DCFW or a DCSG should be configured to fail closed—no traffic should be passed once power has been cut.

6.2.2 DCIPS

Power to the device is cut while passing a mixture of legitimate and disallowed traffic. A DCIPS is expected to fail open; i.e., the IPS engine is bypassed and traffic is allowed through the device. If traffic is not allowed to pass once power is removed, this will be recorded as a failure.

6.3 Persistence of Data

The device should retain all configuration data, policy data, and locally logged data once it has been restored to operation following power failure.

6.4 High Availability (HA) – Optional

High availability (HA) is important to many enterprise customers. This test is designed to evaluate the effectiveness of available HA options. If no HA offering is available, all results in this section will be marked as "NA."

6.4.1 Failover – Legitimate Traffic

Two (or three) identical devices will be configured in an active-passive configuration or active-active N+1 configuration, and legitimate traffic will be passed through the device at 70% of the maximum rated load as determined in section 4.3.2. Switch connectivity or power to the primary device will be terminated, and the device will be expected to failover seamlessly with zero loss of legitimate traffic (some retransmissions are acceptable).

6.4.2 Time to Failover

Time to failover to the standby device will be recorded.

6.4.3 Stateful Operation

This test measures whether full state is maintained across all connections throughout the period of failover.

7 Total Cost of Ownership and Value

Implementation of security solutions can be complex, with several factors affecting the overall cost of deployment, maintenance, and upkeep. All of the following should be considered over the course of the useful life of the solution:

- **Product Purchase** – The cost of acquisition
- **Product Maintenance** – The fees paid to the vendor, including software and hardware support, maintenance, and other updates
- **Installation** – The time required to take the device out of the box, configure it, put it into the network, apply updates and patches, and set up desired logging and reporting
- **Upkeep** – The time required to apply periodic updates and patches from vendors, including hardware, software, and other updates

Appendix A: Change Log

Version 2.0 – 5 October, 2018

- Section 1.1: Edited for clarity
- Section 1.5: Added feedback address
- Section 4.3: Edited for clarity
- Removed Section 4.6 “Real-World” Traffic Mixes
- Added Section 4.6 “Real-World” Single Application Flows
- Section 6.0: Edited for clarity

Contact Information

NSS Labs, Inc.
3711 South Mopac Expressway
Building 1, Suite 400
Austin, TX 78746
info@nsslabs.com
www.nsslabs.com