DATA CENTER INTRUSION PREVENTION SYSTEM TEST REPORT

Fortinet FortiGate 6300F v5.4.10 GA Build 4283

OCTOBER 30, 2018

Authors – Keith Bormann, Ryan Turner, Matt Chips, Matt Wheeler

Overview

NSS Labs performed an independent test of the Fortinet FortiGate 6300F v5.4.10 GA Build 4283. The product was subjected to thorough testing at the NSS facility in Austin, Texas, based on the Data Center Network Security (DCNS) Test Methodology v2.0,[1] available at www.nsslabs.com. This test was conducted free of charge and NSS did not receive any compensation in return for Fortinet's participation.

While the companion Comparative Reports on security, performance, and total cost of ownership (TCO) will provide information about all tested products, this Test Report provides detailed information not available elsewhere.

NSS research indicates that DCIPS devices are typically deployed to protect data center assets, and most enterprises will tune intrusion prevention system (IPS) modules within their DCIPS. Therefore, during NSS testing, DCIPS products are configured with a tuned policy setting in order to provide readers with relevant security effectiveness and performance dimensions based on their expected usage.

Product	Exploit Block Rate[2]	Evasions Blocked	Stability & Reliability	3-Year TCO (US$)
Fortinet FortiGate 6300F v5.4.10 GA Build 4283	99.01%	99/99[3]	PASS	$258,000
	Resiliency	Transactional Use Case	Multimedia Use Case	Corporate Use Case
	77.14%	49,562 Mbps	91,320 Mbps	66,323 Mbps

Figure 1 – Overall Test Results

Using the tuned policy, the Fortinet FortiGate 6300F v5.4.10 GA Build 4283 blocked 99.01% of exploits. The device proved effective against 99 out of 99 evasions it was tested against. The device passed all stability and reliability tests.

To represent different types of traffic seen in a data center, NSS has created three different use cases: transactional, multimedia, and corporate. For each of these weighted use cases, *NSS-Tested Throughput* is calculated by taking an average of the device's IPv4 and IPv6 results. NSS Labs rates the FortiGate 6300F throughput as follows:

- Transactional use case: 49,562 Mbps
- Multimedia use case: 91,320 Mbps
- Corporate use case: 66,323 Mbps

[1] This methodology covers a range of devices that provide network security for the data center, one of which is the data center intrusion prevention system (DCIPS). For more information, visit www.nsslabs.com.

[2] Exploit block rate is defined as a percentage of the total number of exploits that are blocked under test.

[3] In accordance with the industry standard for vulnerability disclosures and to provide vendors with sufficient time to add protection where necessary, NSS Labs will not publicly release information about which previously unpublished techniques were applied during testing until 90 days after the publication of this document.

Table of Contents

Table of Figures

Security Effectiveness

This section verifies that the device can enforce the security policy effectively. Security effectiveness was tested over IPv4 only. For systems that may be exposed to threats over IPv6, NSS recommends that enterprises validate security effectiveness using IPv6.

False Positive Testing

Any signature that blocks non-malicious traffic during false-positive testing is disabled for security testing.

NSS Exploit Library

NSS' security effectiveness testing leverages the deep expertise of our engineers who utilize multiple commercial, open-source, and proprietary tools as appropriate. With more than 2,000 exploits, this is the industry's most comprehensive test to date.

Product	Total Number of Threats Run	Total Number of Threats Blocked	Block Percentage
Fortinet FortiGate 6300F v5.4.10 GA Build 4283	2,223	2,201	99.01%

Figure 2 – Number of Threats Blocked (%)

Resiliency

NSS also measured the resiliency of a device by introducing previously unseen variations of a known exploit and measuring the device's effectiveness against them. Figure 3 depicts the resiliency score.

Product	Block Percentage
Fortinet FortiGate 6300F v5.4.10 GA Build4283	77.14%

Figure 3 – Resiliency Score

Coverage by Impact Type

The most serious exploits are those that result in a remote system compromise, providing the attacker with the ability to execute arbitrary system-level commands. Most exploits in this class are "weaponized" and offer the attacker a fully interactive remote shell on the target server. Slightly less serious are attacks that result in an individual service compromise, but not arbitrary system-level command execution. Finally, there are attacks that result 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. Clients can contact NSS for more information about these tests.

Coverage by Date

Figure 4 provides insight into whether or not a vendor is aging out protection signatures aggressively enough to preserve performance levels. It also reveals whether a product lags behind in protection for the most current vulnerabilities. NSS reports exploits by individual years for the past 10 years. Exploits older than 10 years are grouped together.

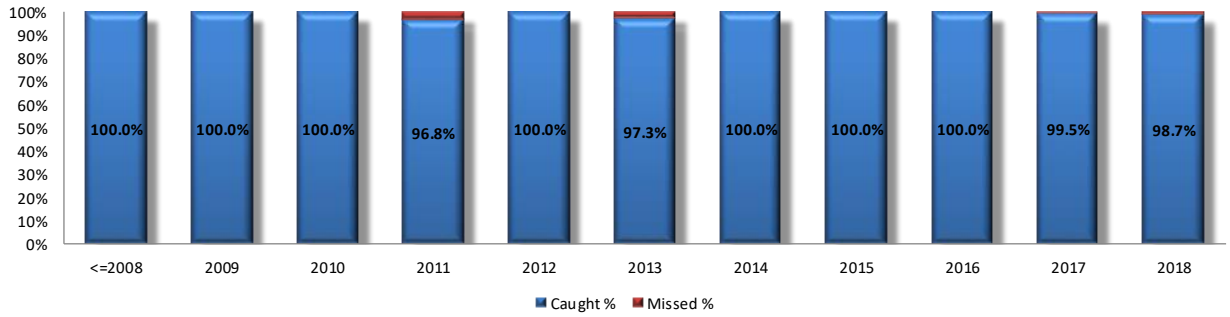


Figure 4 – Product Coverage by Date

Coverage by Target Vendor

Exploits within the *NSS Exploit Library* target a wide range of protocols and applications. Figure 5 depicts the coverage offered for five of the top vendors targeted in this test. Clients can contact NSS for more information.

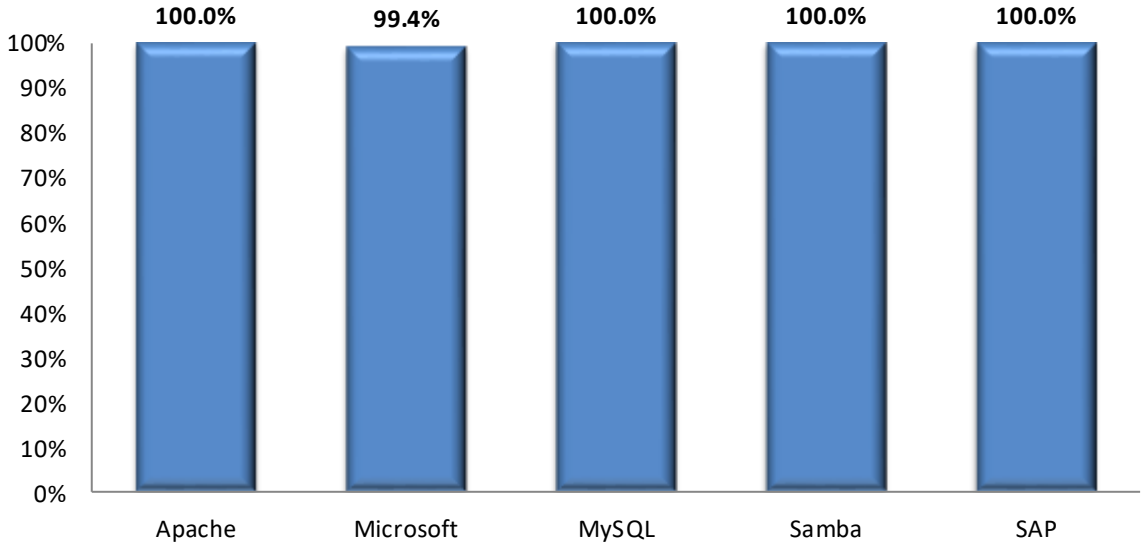


Figure 5 –Product Coverage by Target Vendor

Resistance to Evasion Techniques

Evasion techniques are a means of disguising and modifying attacks at the point of delivery to avoid detection and blocking by security products. Failure of a security device to correctly identify a specific type of evasion potentially allows an attacker to use an entire class of exploits for which the device is assumed to have protection. This renders the device virtually useless. Many of the techniques used in this test have been widely known for years and should be considered minimum requirements for the DCIPS product category.

Providing exploit protection results without fully factoring in evasion can be misleading. The more classes of evasion that are missed (such as IP packet fragmentation, stream segmentation, RPC fragmentation, URL obfuscation, and FTP evasion), the less effective the device. For example, it is better to miss all techniques in one evasion category, such as FTP evasion, than it is to miss one technique in each category, which would result in a broader attack surface.

Furthermore, evasions operating at the lower layers of the network stack (IP packet fragmentation or stream segmentation) have a greater impact on security effectiveness than those operating at the upper layers (URL or FTP obfuscation). Lower-level evasions will potentially impact a wider number of exploits; missing TCP segmentation, for example, is a much more serious issue than missing FTP obfuscation.

The resiliency of a system can be defined as its ability to absorb an attack and reorganize around a threat. When an attacker is presented with a vulnerability, the attacker can select one or more paths to trigger the vulnerability. NSS will measure a device's resiliency by introducing a vulnerability along with its triggers and then asking the device to protect against the vulnerability. NSS will introduce various, previously unseen variations of exploits to exploit the vulnerability and measure the device's effectiveness against them.

A resilient device will be able to detect and prevent against different variations of the exploit. For more, see the Evasions Test Methodology v1.1 at www.nsslabs.com.

Figure 6 provides the results of the evasion tests for the FortiGate 6300F. The FortiGate 6300F blocked all 99 of the evasions it was tested against. For further detail, please reference Appendix A.

Test Procedure	Result
RPC Fragmentation	PASS
URL Obfuscation	PASS
FTP/Telnet Evasion	PASS
IP Packet Fragmentation + TCP Segmentation	PASS
Resiliency[4]	
Attacks on nonstandard ports[5]	PASS

Figure 6 – Resistance to Evasion Results

[4] The results of resiliency testing are included in the Exploit Block Rate calculations.

[5] Enterprises should be aware of the importance of performing deep packet inspection on all packets and ports and over all protocols in order to secure applications effectively.

Performance

There is frequently a trade-off between security effectiveness and performance. Because of this trade-off, it is important to judge a product's security effectiveness within the context of its performance and vice versa. This ensures that new security protections do not adversely impact performance and that security shortcuts are not taken to maintain or improve performance. Performance was tested over IPv4 and IPv6 protocols for all tests except the UDP tests, where performance was tested only over the IPv4 protocol.

In addition, when considering a security device (e.g., an IPS) for the data center rather than for the network perimeter, there are several key metrics that must be adjusted. Performance metrics, while important in any security device, become critical in a device that is intended for data center deployment. In a data center, the volume of traffic is significantly higher than it would be for a device that is intended to protect end user desktops behind the corporate network perimeter. A data center security device also needs to support much higher data rates as it handles traffic for potentially hundreds of thousands of users who are accessing large applications in a server farm inside the network perimeter. Connection rate and concurrent connection capacity are additional metrics that become even more important in a data center security device.

The mix of traffic will differ significantly between a corporate network perimeter and a data center, and this can put additional load on the IPS inspection process. Stateless UDP traffic (such as that seen in a network file system [NFS]) and long-lived transmission control protocol (TCP) connections {as would be seen in an iSCSI storage area network [SAN] or backup application) are common in many data center networks. These types of applications present a continuous and heavy load for the network.

Within the data center, application traffic puts a very different load on the network than does file system traffic. Communications between users and servers have different profiles than do communications between applications, databases, and directory servers. Application traffic is connection-intensive, with connections constantly being set up and torn down. A DCIPS that includes any application awareness capabilities will find significant challenges in data center deployments. Another critical concern is latency, since applications will be adversely affected if the DCIPS introduces delays.

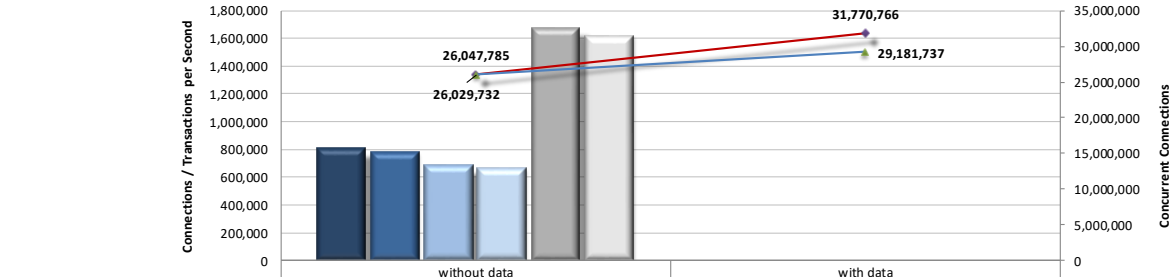
Maximum Capacity

The use of traffic generation appliances allows NSS engineers to create "real-world" traffic at multi-Gigabit speeds as a background load for the tests. The aim of these tests is to stress the inspection engine and determine how it copes with high volumes of TCP connections per second, application layer transactions per second, and concurrent open connections. All packets contain valid payload and address data, and these tests provide an excellent representation of a live network at various connection/transaction rates.

Note that in all tests the following critical "breaking points"—where the final measurements are taken—are used:

- **Excessive concurrent TCP connections** – Latency within the device is causing an unacceptable increase in open connections.
- **Excessive concurrent HTTP connections** – Latency within the device is causing excessive delays and increased response time.
- **Unsuccessful HTTP transactions** – Normally, there should be zero unsuccessful transactions. Once these appear, it is an indication that excessive latency within the device is causing connections to time out.

Figure 7 depicts the results of the IPv4 and IPv6 tests for maximum capacity.



	without data	with data
Max. TCP CPS (IPv4)	804,000	
Max. TCP CPS (IPv6)	780,200	
Max. HTTP CPS (IPv4)	683,900	
Max. HTTP CPS (IPv6)	659,800	
Max. HTTP TPS (IPv4)	1,676,000	
Max. HTTP TPS (IPv6)	1,618,000	
Max. Concurrent TCP Connections (IPv4)	26,047,785	31,770,766
Max. Concurrent TCP Connections (IPv6)	26,029,732	29,181,737

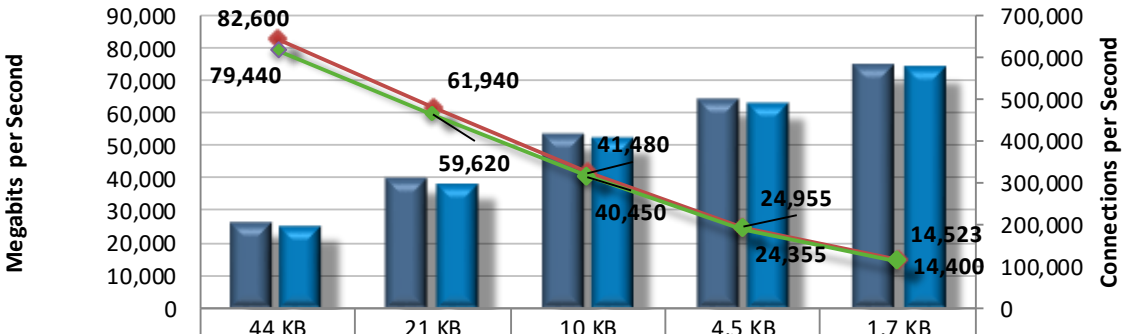
Figure 7 – Concurrency and Connection Rates (IPv4 and IPv6)

HTTP Capacity

The aim of this test 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 possible, while ensuring absolute accuracy and repeatability.

Each transaction consists of a single HTTP GET request. All packets contain valid payload (a mix of binary and ASCII objects) and address data. This test provides an excellent representation of a live network (albeit one biased toward HTTP traffic) at various network loads.

Figure 8 depicts the results of the IPv4 and IPv6 tests for HTTP capacity with no transaction delays.



	44 KB Response	21 KB Response	10 KB Response	4.5 KB Response	1.7 KB Response
IPv4 CPS	206,500	309,700	414,800	499,100	580,900
IPv6 CPS	198,600	298,100	404,500	487,100	576,000
IPv4 (Mbps)	82,600	61,940	41,480	24,955	14,523
IPv6 (Mbps)	79,440	59,620	40,450	24,355	14,400

Figure 8 – HTTP Capacity with No Transaction Delays

Application Average Response Time – HTTP

Application Average Response Time – HTTP (at 95% Maximum Load)	IPv4 Results	IPv6 Results
2,500 Connections per Second – 44 KB Response	3.15	3.18
5,000 Connections per Second – 21 KB Response	2.41	2.40
10,000 Connections per Second – 10 KB Response	1.96	1.90
20,000 Connections per Second – 4.5 KB Response	1.43	1.34
40,000 Connections per Second – 1.7 KB Response	1.48	1.48

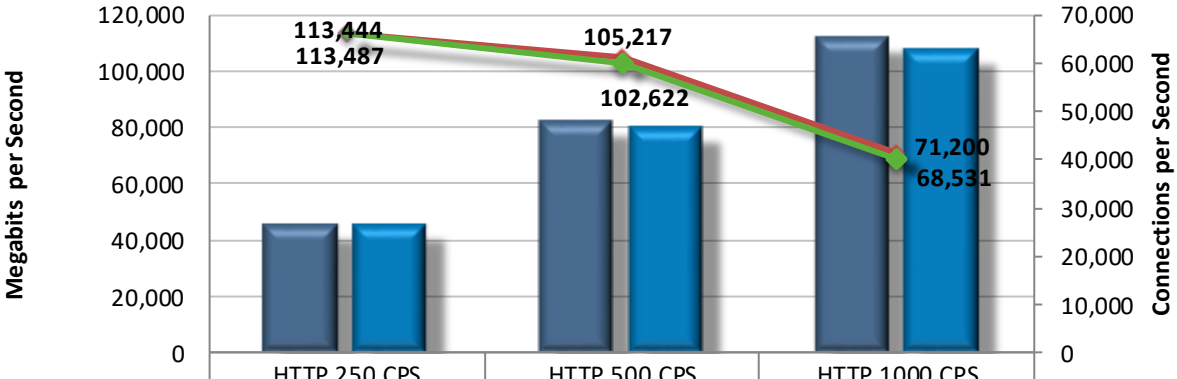
Figure 9 – Average Application Response Time (Milliseconds)

HTTP Capacity with HTTP Persistent Connections

The aim of this test is to determine how the DCIPS copes with network loads of varying average packet size and varying connections per second while inspecting traffic. By creating genuine session-based traffic with varying session lengths, the DCIPS 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.

Figure 10 depicts the results of the IPv4 and IPv6 tests for HTTP capacity with HTTP persistent connections.

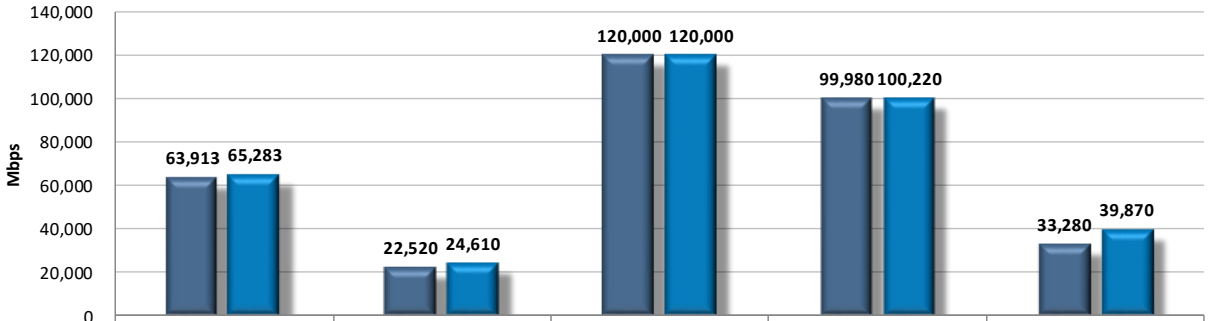


	HTTP 250 CPS	HTTP 500 CPS	HTTP 1000 CPS
IPv4 CPS	26,510	48,250	65,350
IPv6 CPS	26,500	47,060	62,900
IPv4 (Mbps)	113,487	105,217	71,200
IPv6 (Mbps)	113,444	102,622	68,531

Figure 10 – HTTP Capacity with HTTP Persistent Connections

Single Application Flows

This test measures the performance of the device with single application flows. For details about single application flow testing, see the NSS Labs Data Center Network Security (DCNS) Test Methodology v2.0, available at www.nsslabs.com.



	DB	Financial	File Share	Video	Email
IPv4	63,913	22,520	120,000	99,980	33,280
IPv6	65,283	24,610	120,000	100,220	39,870

Figure 11 — Single Application Flows

Raw Packet Processing Performance (UDP Throughput)

This test uses UDP packets of varying sizes generated by test equipment. A constant stream of the appropriate packet size, with variable source and destination IP addresses transmitting from a fixed source port to a fixed destination port, is transmitted bidirectionally 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. Multiple tests are run and averages are taken where necessary.

This traffic does not attempt to simulate any form of a "real-world" network condition. No TCP sessions are created during this test, and there is very little for the detection engine to do. However, each vendor is required to write a signature to detect the test packets in order to ensure that they are being passed through the detection engine and are not being "fast-pathed."

The aim of this test is to determine the raw packet processing capability of each inline port pair of the device, and to determine the device's effectiveness at forwarding packets quickly in order to provide the highest level of network performance with the lowest amount of latency.

Figure 12 depicts the results of the IPv4 tests for raw packet processing performance.

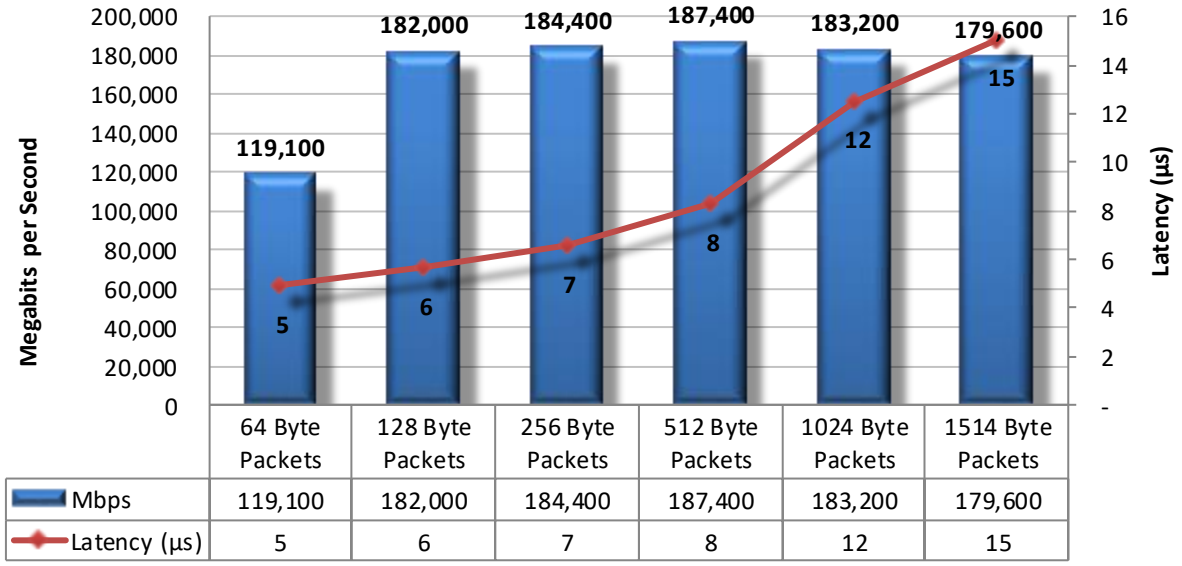


Figure 12 – Raw Packet Processing Performance – UDP Traffic (IPv4)

Raw Packet Processing Performance (UDP Latency)

DCIPS that introduce high levels of latency lead to unacceptable response times for users, especially where multiple security devices are placed in the data path. Figure 13 depicts UDP latency (in microseconds) as recorded during the UDP throughput tests at 95% of the maximum load for IPv4.

Latency – UDP	IPv4 Results
64-Byte Packets	5
128-Byte Packets	6
256-Byte Packets	7
512-Byte Packets	8
1024-Byte Packets	12
1514-Byte Packets	15

Figure 13 – UDP Latency in Microseconds

NSS-Tested Throughput: Use Cases

Because data center network traffic can vary greatly between industries and enterprises, NSS has created three separate use cases. Each use case weights test results in order to align them with the different use cases seen in a data center, i.e., transactional, multimedia or corporate.

The transactional use case is intended to represent a data center with traffic that is more transactional in nature. An example of this may include B2B (business-to-business) or B2C (business-to-consumer) e-commerce. The rated throughput emphasizes smaller packet sizes and connections per second.

The multimedia use case is intended to represent a data center whose purpose is to serve media content. The rated throughput emphasizes larger packet sizes, maximum concurrent sessions, and streaming protocols.

The corporate use case may be best described as the data center footprint of a typical enterprise, where mission-critical applications such as email and ERP (enterprise resource planning software) are kept. The rated throughput emphasizes various packet sizes and protocols that are more likely to be found in those situations, such as email, database, and file sharing.

Use Case	IPv4	IPv6	Results
Transactional (small packets, database, email)	50,267	48,858	49,562
Multimedia (video, large packets, database, email)	91,757	90,882	91,320
Corporate (email, file share, database, mix of packet sizes)	67,125	65,521	66,323

Figure 14 – NSS-Tested Throughput Use Cases

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 passing malicious traffic. Products that cannot sustain legitimate traffic (or that crash) while under hostile attack will not pass. Stability and reliability was tested over IPv4 only.

The device is required to remain operational and stable throughout these tests, and to block 100% of previously blocked traffic, raising an alert for each. If any non-allowed traffic passes successfully, caused either by the volume of traffic or by the device failing open for any reason, the device will fail the test.

Stability and Reliability	Result
Attack Detection/Blocking – Normal Load	PASS
State Preservation – Normal Load	PASS
Pass Legitimate Traffic – Normal Load	PASS
State Preservation – Maximum Exceeded	PASS

Figure 15 – Stability and Reliability Results

These tests also determine the behavior of the state engine under load. All DCIPS devices must choose whether to risk denying legitimate traffic or risk allowing malicious traffic once they run low on resources. A DCIPS device will drop new connections when resources (such as state table memory) are low, or when traffic loads exceed its capacity. In theory, this means the DCIPS will block legitimate traffic but maintain state on existing connections (and prevent attack leakage).

Total Cost of Ownership (TCO)

Implementation of security solutions can be complex, with several factors affecting the overall cost of deployment, maintenance, and upkeep. Each 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
- **Management** – Day-to-day management tasks, including device configuration, policy updates, policy deployment, alert handling, and so on

For the purposes of this report, capital expenditure (capex) items are included for a single device only (the cost of acquisition and installation).

Installation Hours

Figure 16 depicts the number of hours of labor required to install each device using only local device management options. The table accurately reflects the amount of time that NSS engineers, with the help of vendor engineers, needed to install and configure the device to the point where it operated successfully in the test harness, passed legitimate traffic, and blocked and detected prohibited or malicious traffic. This closely mimics a typical enterprise deployment scenario for a single device.

Installation cost is based on the time that an experienced security engineer would require to perform the installation tasks described above. This approach allows NSS to hold constant the talent cost and measure only the difference in time required for installation. Readers should substitute their own costs to obtain accurate TCO figures.

Product	Installation (Hours)
Fortinet FortiGate 6300F v5.4.10 GA Build 4283	8

Figure 16 – Device Installation Time (Hours)

Total Cost of Ownership

Calculations are based on vendor-provided pricing information. Where possible, the 24/7 maintenance and support option with 24-hour replacement is utilized, since this is the option typically selected by enterprise customers. Prices are for single device management and maintenance only; costs for central management solutions (CMS) may be extra.

Product	Purchase Price	Maintenance/ Year	Year 1 Cost	Year 2 Cost	Year 3 Cost	3-Year TCO
Fortinet FortiGate 6300F v5.4.10 GA Build 4283	$117,000	$46,800	$164,400	$46,800	$46,800	$258,000

Figure 17 –3-Year TCO (US$)

- **Year 1 Cost** is calculated by adding installation costs (US$75 per hour fully loaded labor x installation time) + purchase price + first-year maintenance/support fees.
- **Year 2 Cost** consists only of maintenance/support fees.
- **Year 3 Cost** consists only of maintenance/support fees.

For additional TCO analysis, including for the CMS, refer to the TCO Comparative Report.

Appendix A: Product Scorecard

Security Effectiveness	
Block Rate	99.01%
False Positive Testing	PASS
Evasions and Attack Leakage	
IP Packet Fragmentation	
(overlapping small IP fragments favoring new data)	PASS
(overlapping small IP fragments favoring new data in reverse order)	PASS
(overlapping small IP fragments favoring new data in random order)	PASS
(overlapping small IP fragments favoring new data; interleave chaff (invalid IP options))	PASS
(overlapping small IP fragments favoring new data in random order; interleave chaff (invalid IP options))	PASS
(overlapping small IP fragments favoring new data in random order; interleave chaff (invalid IP options); delay random fragment)	PASS
(overlapping small IP fragments favoring new data; interleave chaff (invalid IP options); DSCP value 16)	PASS
(overlapping small IP fragments favoring new data in random order; interleave chaff (invalid IP options); delay random fragment; DSCP value 34)	PASS
(small IP fragments)	PASS
(small IP fragments in reverse order)	PASS
(small IP fragments in random order)	PASS
(small IP fragments; delay first fragment)	PASS
(small IP fragments in reverse order; delay last fragment)	PASS
(small IP fragments; interleave chaff (invalid IP options))	PASS
(small IP fragments in random order; interleave chaff (invalid IP options))	PASS
(small IP fragments in random order; interleave chaff (invalid IP options); delay random fragment)	PASS
(small IP fragments; interleave chaff (invalid IP options); DSCP value 16)	PASS
(small IP fragments in random order; interleave chaff (invalid IP options); delay random fragment; DSCP value 34)	PASS
(overlapping small TCP segments favoring new data)	PASS
(overlapping small TCP segments favoring new data in reverse order)	PASS
(overlapping small TCP segments favoring new data in random order)	PASS
(overlapping small TCP segments favoring new data; delay first segment)	PASS
(overlapping small TCP segments favoring new data in reverse order; delay last segment)	PASS
(overlapping small TCP segments favoring new data; interleave chaff (invalid TCP checksums); delay first segment)	PASS
(overlapping small TCP segments favoring new data in random order; interleave chaff (older PAWS timestamps); delay last segment)	PASS
(overlapping small TCP segments favoring new data in random order; interleave chaff (out-of-window sequence numbers); TCP MSS option)	PASS
(overlapping small TCP segments favoring new data in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option)	PASS
(overlapping small TCP segments favoring new data in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option; delay first segment)	PASS
(small TCP segments)	PASS
(small TCP segments in reverse order)	PASS
(small TCP segments in random order)	PASS
(small TCP segments; delay first segment)	PASS
(small TCP segments in reverse order; delay last segment)	PASS

(small TCP segments; interleave chaff (invalid TCP checksums); delay first segment)	PASS
(small TCP segments in random order; interleave chaff (older PAWS timestamps); delay last segment)	PASS
(small TCP segments in random order; interleave chaff (out-of-window sequence numbers); TCP MSS option)	PASS
(small TCP segments in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option)	PASS
(small TCP segments in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option; delay first segment)	PASS
(overlapping small TCP segments favoring new data; small IP fragments)	PASS
(small TCP segments; overlapping small IP fragments favoring new data)	PASS
(overlapping small TCP segments favoring new data; overlapping small IP fragments favoring new data)	PASS
(overlapping small TCP segments favoring new data in random order; small IP fragments in random order)	PASS
(small TCP segments in random order; overlapping small IP fragments favoring new data in random order)	PASS
(overlapping small TCP segments favoring new data in random order; overlapping small IP fragments favoring new data in random order)	PASS
(overlapping small TCP segments favoring new data in random order; overlapping small IP fragments favoring new data in random order; interleave chaff (invalid IP options))	PASS
(overlapping small TCP segments favoring new data; interleave chaff (invalid TCP checksums); small IP fragments; interleave chaff (invalid IP options))	PASS
(small TCP segments; interleave chaff (invalid TCP checksums); overlapping small IP fragments favoring new data; interleave chaff (invalid IP options))	PASS
(small TCP segments; interleave chaff (invalid TCP checksums); delay last segment; overlapping small IP fragments favoring new data; interleave chaff (invalid IP options))	PASS
(small TCP segments; small IP fragments)	PASS
(small TCP segments; small IP fragments in reverse order)	PASS
(small TCP segments in random order; small IP fragments)	PASS
(small TCP segments; small IP fragments in random order)	PASS
(small TCP segments in random order; small IP fragments in reverse order)	PASS
(small TCP segments in random order; interleave chaff (invalid TCP checksums); small IP fragments in reverse order; interleave chaff (invalid IP options))	PASS
(small TCP segments; interleave chaff (invalid TCP checksums); delay last segment; small IP fragments; interleave chaff (invalid IP options))	PASS
(small TCP segments; interleave chaff (invalid TCP checksums); small IP fragments; interleave chaff (invalid IP options); delay last fragment)	PASS
(small TCP segments in random order; interleave chaff (out-of-window sequence numbers); TCP MSS option; small IP fragments in random order; interleave chaff (invalid IP options); delay random fragment)	PASS
(small TCP segments in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option; delay first segment; small IP fragments)	PASS
Resiliency	
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS

Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	FAIL
Attacks on nonstandard ports	PASS
RPC Fragmentation	
One-byte fragmentation (ONC)	PASS
Two-byte fragmentation (ONC)	PASS
All fragments, including Last Fragment (LF) will be sent in one TCP segment (ONC)	PASS
All frags except Last Fragment (LF) will be sent in one TCP segment. LF will be sent in separate TCP seg (ONC)	PASS
One RPC fragment will be sent per TCP segment (ONC)	PASS
One LF split over more than one TCP segment. In this case no RPC fragmentation is performed (ONC)	PASS
Canvas Reference Implementation Level 1 (MS)	PASS
Canvas Reference Implementation Level 2 (MS)	PASS
Canvas Reference Implementation Level 3 (MS)	PASS
Canvas Reference Implementation Level 4 (MS)	PASS
Canvas Reference Implementation Level 5 (MS)	PASS
Canvas Reference Implementation Level 6 (MS)	PASS
Canvas Reference Implementation Level 7 (MS)	PASS
Canvas Reference Implementation Level 8 (MS)	PASS
Canvas Reference Implementation Level 9 (MS)	PASS
Canvas Reference Implementation Level 10 (MS)	PASS
URL Obfuscation	
URL encoding - Level 1 (minimal)	PASS

URL encoding - Level 2		PASS
URL encoding - Level 3		PASS
URL encoding - Level 4		PASS
URL encoding - Level 5		PASS
URL encoding - Level 6		PASS
URL encoding - Level 7		PASS
URL encoding - Level 8 (extreme)		PASS
Directory Insertion		PASS
Premature URL ending		PASS
Long URL		PASS
Fake parameter		PASS
TAB separation		PASS
Case sensitivity		PASS
Windows \ delimiter		PASS
Session splicing		PASS
FTP/Telnet Evasion		
Inserting spaces in FTP command lines		PASS
Inserting non-text Telnet opcodes - Level 1 (minimal)		PASS
Inserting non-text Telnet opcodes - Level 2		PASS
Inserting non-text Telnet opcodes - Level 3		PASS
Inserting non-text Telnet opcodes - Level 4		PASS
Inserting non-text Telnet opcodes - Level 5		PASS
Inserting non-text Telnet opcodes - Level 6		PASS
Inserting non-text Telnet opcodes - Level 7		PASS
Inserting non-text Telnet opcodes - Level 8 (extreme)		PASS
Performance	IPv4	IPv6
Raw Packet Processing Performance (UDP Traffic)	Mbps	Mbps
64 Byte Packets	119,100	NA
128 Byte Packets	182,000	NA
256 Byte Packets	184,400	NA
512 Byte Packets	187,400	NA
1024 Byte Packets	183,200	NA
1514 Byte Packets	179,600	NA
Latency - UDP	Microseconds	Microseconds
64 Byte Packets	4.89	NA
128 Byte Packets	5.64	NA
256 Byte Packets	6.54	NA
512 Byte Packets	8.29	NA
1024 Byte Packets	12.49	NA
1514 Byte Packets	15.05	NA
Maximum Capacity	CPS	CPS
Theoretical Max. Concurrent TCP Connections	26,047,785	26,029,732
Theoretical Max. Concurrent TCP Connections w/Data	31,770,766	29,181,737

Maximum TCP Connections Per Second	804,000	780,200
Maximum HTTP Connections Per Second	683,900	659,800
Maximum HTTP Transactions Per Second	1,676,000	1,618,000
HTTP Capacity With No Transaction Delays	CPS	CPS
25,000 Connections Per Second – 44Kbyte Response	206,500	198,600
50,000 Connections Per Second – 21Kbyte Response	309,700	298,100
100,000 Connections Per Second – 10Kbyte Response	414,800	404,500
200,000 Connections Per Second – 4.5Kbyte Response	499,100	487,100
400,000 Connections Per Second – 1.7Kbyte Response	580,900	576,000
Application Average Response Time - HTTP (at 95% Max Load)	Milliseconds	Milliseconds
25,000 Connections Per Second – 44Kbyte Response	3.15	3.18
50,000 Connections Per Second – 21Kbyte Response	2.41	2.40
100,000 Connections Per Second – 10Kbyte Response	1.96	1.90
200,000 Connections Per Second – 4.5Kbyte Response	1.43	1.34
400,000 Connections Per Second – 1.7Kbyte Response	1.48	1.48
HTTP Capacity with HTTP Persistent Connections	CPS	CPS
250 Connections per Second	26,510	26,500
500 Connections per Second	48,250	47,060
1000 Connections per Second	65,350	62,900
“Real-World” Traffic	Mbps	Mbps
DB	63,913	65,283
Financial	22,520	24,610
File Share	120,000	120,000
Video	99,980	100,220
Email	33,280	39,870
Stability & Reliability		
Blocking Under Extended Load with Attacks		PASS
Behavior Of The State Engine Under Load		PASS
State Preservation - Normal Load		PASS
State Preservation - Maximum Exceeded		PASS
Total Cost of Ownership		TCO
Ease of Use		
Initial Setup (Hours)		8
Time Required for Upkeep (Hours per Year)		Contact NSS Labs
Time Required to Tune (Hours per Year)		Contact NSS Labs
Expected Costs		
Initial Purchase (hardware as tested)		$117,000
Installation Labor Cost (@$75/hr)		$600
Annual Cost of Maintenance & Support (hardware/software)		$46,800
Annual Cost of Updates (IPS/AV/etc.)		$0
Initial Purchase (centralized management system)		Contact NSS Labs
Annual Cost of Maintenance & Support (centralized management system)		Contact NSS Labs
Management Labor Cost (per Year @$75/hr)		Contact NSS Labs

Tuning Labor Cost (per Year @$75/hr)	Contact NSS Labs
Total Cost of Ownership	
Year 1	$164,400
Year 2	$46,800
Year 3	$46,800
3-Year Total Cost of Ownership	$258,000

Figure 18 – Detailed Scorecard

Test Methodology

Data Center Network Security (DCNS) Test Methodology v2.0

Evasions Test Methodology v1.1

Copies of the test methodologies are available at www.nsslabs.com.

Contact Information

NSS Labs, Inc.
3711 South MoPac Expressway
Building 1, Suite 400
Austin, TX 78746-8022
USA
info@nsslabs.com
www.nsslabs.com

This and other related documents are available at: **www.nsslabs.com**. To receive a licensed copy or report misuse, please contact NSS Labs.