DATA CENTER SECURITY GATEWAY TEST REPORT

Fortinet FortiGate 3000D v5.4.5 GA Build 3273

DECEMBER 21, 2017

Authors – Keith Bormann and Matthew Chips

Overview

NSS Labs performed an independent test of the Fortinet FortiGate 3000D v5.4.5 GA Build 3273 (IPS Engine 3.00437 and IPS DB 12.00236). The product was subjected to thorough testing at the NSS facility in Austin, Texas, based on the Data Center Security Gateway (DCSG) Test Methodology v1.0, which is 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 DCSG devices are typically deployed to protect data center assets, and most enterprises will tune intrusion prevention system (IPS) modules within their DCSGs. Therefore, during NSS testing, DCSG 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	NSS Exploit Block Rate[1]	NSS-Tested Throughput (IPv4)	NSS-Tested Throughput (IPv6)	3-Year TCO (US$)
Fortinet FortiGate 3000D v5.4.5 GA Build 3273	97.97%	30,987 Mbps	30,046 Mbps	$85,209
	Evasions Blocked	Firewall Policy Enforcement		Stability and Reliability
	113/113	PASS		PASS

Figure 1 – Overall Test Results

Using the tuned policy, the FortiGate 3000D blocked 97.97% of exploits. The device proved effective against all evasion techniques tested. The device passed all stability and reliability tests.

The FortiGate 3000D is rated by NSS at 30,987 Mbps for IPv4, which is higher than the vendor-claimed performance, and 30,046 Mbps for IPv6, which is also higher than the vendor-claimed performance; Fortinet rates this device at 23 Gbps. NSS-Tested Throughput is calculated as an average of all the "real-world" protocol mixes and the 21 KB HTTP response-based capacity test.

[1] Exploit block rate is defined as the number of exploits from the *NSS Exploit Library* blocked under test.

Table of Contents

Overview 2

Security Effectiveness 5

Firewall Policy Enforcement 5
False Positive Testing 5
NSS Exploit Library 6
Coverage by Impact Type 6
Coverage by Date 6
Coverage by Target Vendor 7
Resistance to Evasion Techniques 7

Performance 9

Raw Packet Processing Performance (UDP Throughput) 9
Raw Packet Processing Performance (UDP Latency) 10
Maximum Capacity 11
HTTP Capacity 12
Application Average Response Time – HTTP 13
HTTP Capacity with HTTP Persistent Connections 13
Real-World Traffic Mixes 14

Stability and Reliability 15

Total Cost of Ownership (TCO) 16

Installation Hours 16
Total Cost of Ownership 17

Appendix A: Product Scorecard 18

Test Methodology 25

Contact Information 25

Table of Figures

Figure 1 – Overall Test Results....2
Figure 2 – Firewall Policy Enforcement....5
Figure 3 – Number of Exploits Blocked (%)....6
Figure 4 – Product Coverage by Date....6
Figure 5 – Product Coverage by Target Vendor....7
Figure 6 – Resistance to Evasion Results....8
Figure 7 – Raw Packet Processing Performance – UDP Traffic (IPv4)....10
Figure 8 – UDP Latency in Microseconds(IPv4)....10
Figure 9 – Concurrency and Connection Rates (IPv4)....11
Figure 10 – Concurrency and Connection Rates (IPv6)....12
Figure 11 – HTTP Capacity with No Transaction Delays....12
Figure 12 – Average Application Response Time (Milliseconds)....13
Figure 13 – HTTP Capacity with HTTP Persistent Connections....13
Figure 14 – "Real-World" Traffic Mixes....14
Figure 15 – Stability and Reliability Results....15
Figure 16 – Sensor Installation Time (Hours)....16
Figure 17 –3-Year TCO (US$)....17
Figure 18 – Detailed Scorecard....24

Security Effectiveness

This section verifies that the device can enforce the security policy effectively. *Security Effectiveness* was tested over IPv4 only.

Firewall Policy Enforcement

NSS requires one security policy to be applied to all interfaces under test. Policies are rules 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. At a minimum, the firewall must provide a trusted internal interface and an untrusted external/Internet interface. Policies are typically written to permit or deny network traffic from one or both of the following zones:

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

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

- Trusted to Untrusted
- Untrusted to Trusted

Test Procedure	Result
Baseline Policy	PASS
Simple Policy	PASS
Complex Policy	PASS
Static NAT	PASS
SYN Flood Protection	PASS
IP Address Spoofing Protection	PASS

Figure 2 – Firewall Policy Enforcement

False Positive Testing

The FortiGate 3000D correctly identified traffic and did not fire alerts for non-malicious content.

NSS Exploit Library

NSS' security effectiveness testing leverages the deep expertise of our engineers who utilize multiple commercial, open-source, and proprietary tools, including NSS' network live stack test environment as appropriate. With 1,974 exploits, this is the industry's most comprehensive test to date. Most notably, the exploits and payloads in this test have been validated such that:

- 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
- Etc.

Product	Total Number of Exploits Run	Total Number Blocked	Block Percentage
Fortinet FortiGate 3000D v5.4.5 GA Build 3273	1,974	1,934	97.97%

Figure 3 – Number of Exploits Blocked (%)

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 ten years. Exploits older than ten years are grouped together.

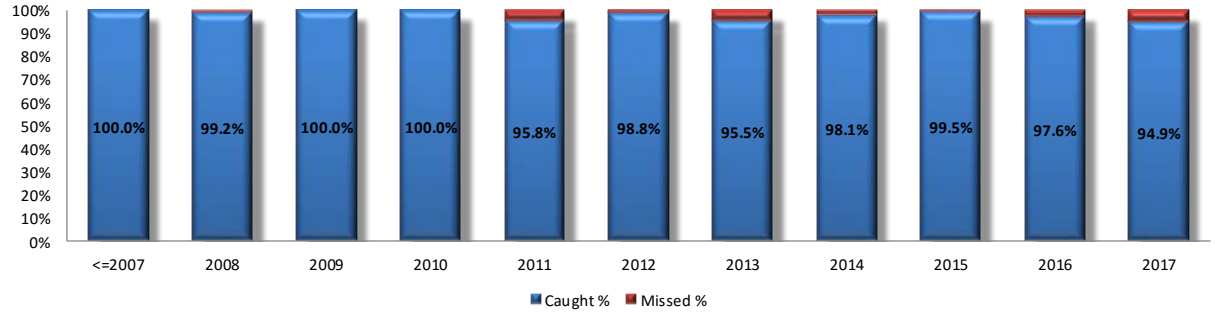


Figure 4 – Product Coverage by Date

Coverage by Target Vendor

Exploits within the *NSS Exploit Library* target a wide range of protocols and applications. Figure 5 depicts the coverage offered by the FortiGate 3000D for five of the top vendors targeted in this test. Clients can contact NSS for more information about this test.

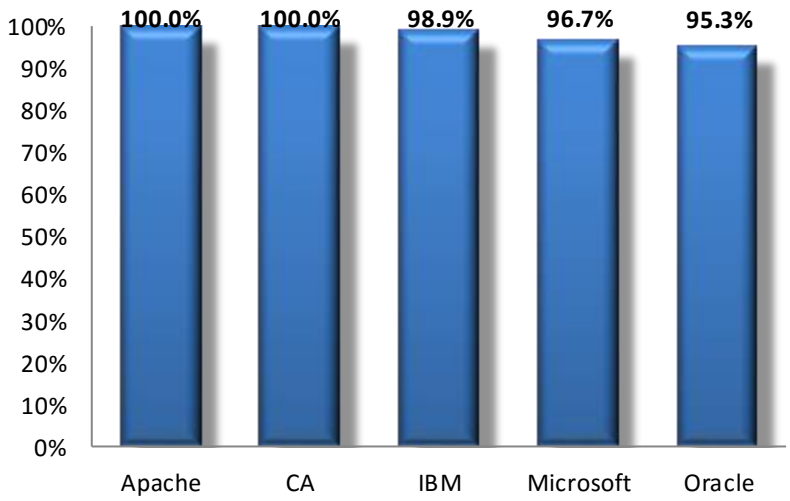


Figure 5 – Product Coverage by Target Vendor

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 DCSG 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 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.

Figure 6 provides the results of the evasion tests for the FortiGate 3000D. The FortiGate 3000D blocked all of the 113 evasions it was tested against. For further detail, please reference Appendix A.

Test Procedure	Result
IP Packet Fragmentation	PASS
TCP Stream Segmentation	PASS
RPC Fragmentation	PASS
URL Obfuscation	PASS
FTP/Telnet Evasion	PASS
IP Packet Fragmentation + TCP Segmentation	PASS
IP Fragmentation + MSRPC Fragmentation	PASS
IP Fragmentation + SMB Evasions	PASS
TCP Segmentation + SMB / NetBIOS Evasions	PASS
IP Fragmentation + MSRPC Fragmentation	PASS

Figure 6 – Resistance to Evasion Results

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 only tested 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.

Traffic mix 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 to the network.

Within the data center, application traffic puts a very different load on the network than file system traffic does. Communications between users and servers have very different profiles than communications between applications, databases, and directory servers. Application traffic is connection-intensive, with connections constantly being set up and torn down. A DCSG 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 DCSG introduces delays.

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 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 least amount of latency. Figure 7 depicts the results of the IPv4 tests for raw packet processing performance.

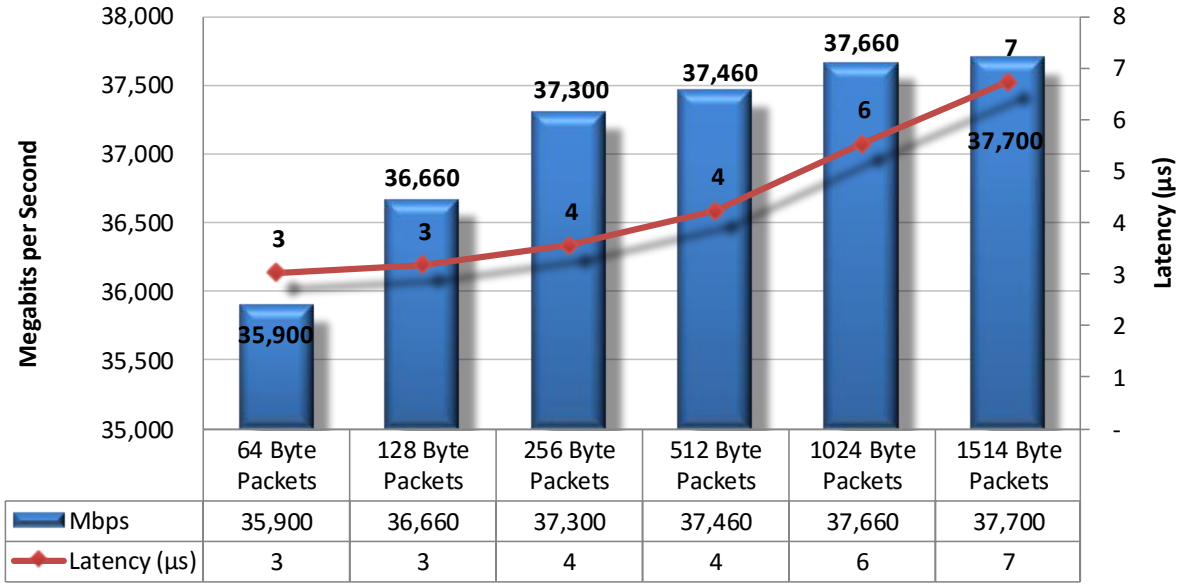


Figure 7 – Raw Packet Processing Performance – UDP Traffic (IPv4)

Raw Packet Processing Performance (UDP Latency)

DCSGs 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 8 depicts UDP latency (in microseconds) as recorded during the UDP throughput tests at 80% of maximum load for the IPv4.

Latency – UDP	IPv4 Results
64-Byte Packets	3.03
128-Byte Packets	3.17
256-Byte Packets	3.56
512-Byte Packets	4.23
1024-Byte Packets	5.52
1514-Byte Packets	6.73

Figure 8 – UDP Latency in Microseconds(IPv4)

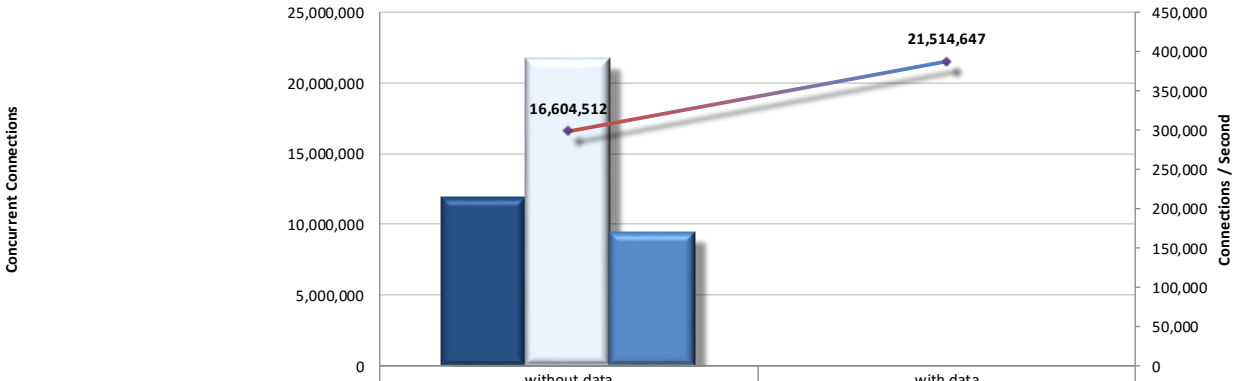
Maximum Capacity

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 DCSG is causing an unacceptable increase in open connections.
- **Excessive concurrent HTTP connections** – Latency within the DCSG 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 DCSG is causing connections to time out.

Figure 9 and Figure 10 depict the results of the IPv4 and IPv6 tests for maximum capacity.



	without data	with data
Maximum TCP Connections per Second	216,000	
Maximum HTTP Transactions per Second	390,300	
Maximum HTTP Connections per Second	170,000	
Theoretical Max. Concurrent TCP Connections	16,604,512	21,514,647

Figure 9 – Concurrency and Connection Rates (IPv4)

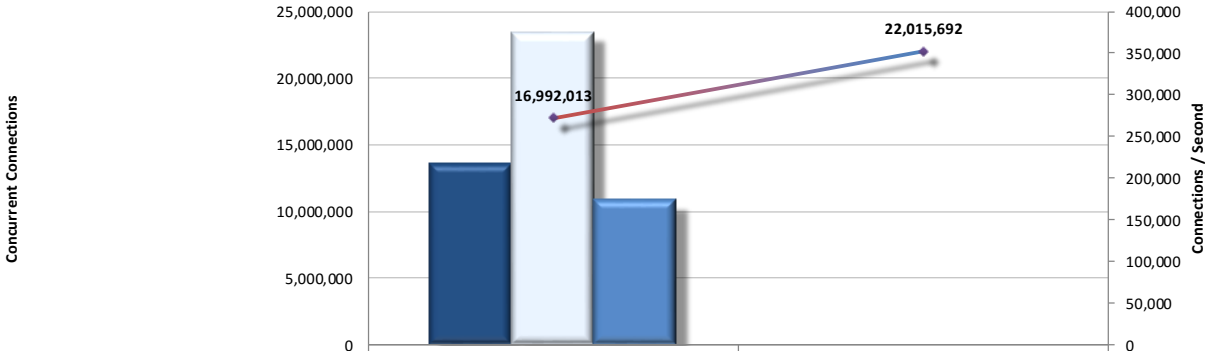


Figure 10 – Concurrency and Connection Rates (IPv6)

HTTP Capacity

The aim of the HTTP capacity 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 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 11 depicts the results of the IPv4 and IPv6 tests for HTTP capacity with no transaction delays.

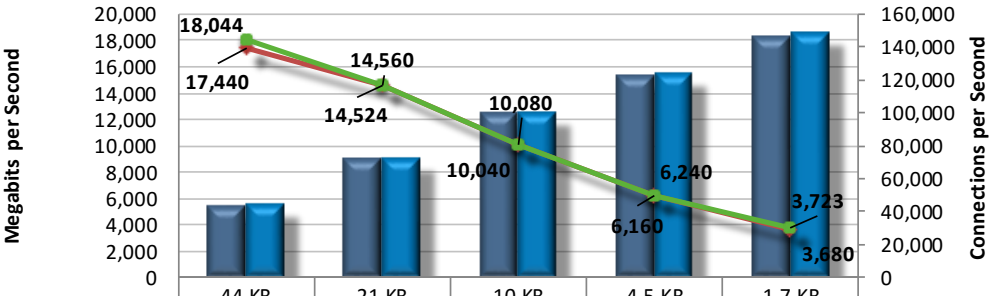


Figure 11 – HTTP Capacity with No Transaction Delays

Application Average Response Time – HTTP

Application Average Response Time – HTTP (at 90% Maximum Load)	IPv4 Results	IPv6 Results
2,500 Connections per Second – 44 KB Response	2.84	3.26
5,000 Connections per Second – 21 KB Response	3.15	3.22
10,000 Connections per Second – 10 KB Response	3.02	2.86
20,000 Connections per Second – 4.5 KB Response	2.51	2.38
40,000 Connections per Second – 1.7 KB Response	3.35	2.95

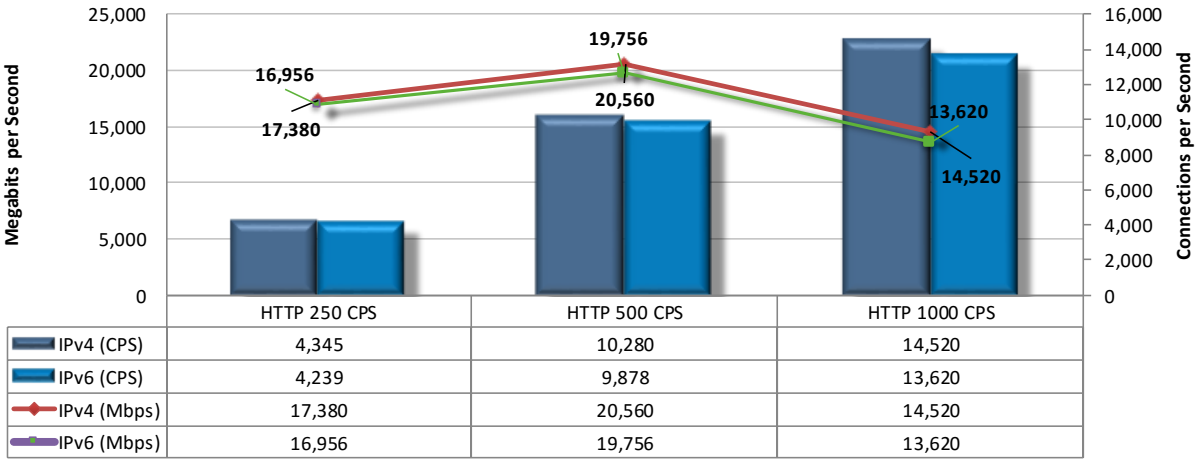
Figure 12 – Average Application Response Time (Milliseconds)

HTTP Capacity with HTTP Persistent Connections

The aim of these tests is to determine how the DCSG 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 DCSG 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 13 depicts the results of the IPv4 and IPv6 tests for HTTP capacity with HTTP persistent connections.

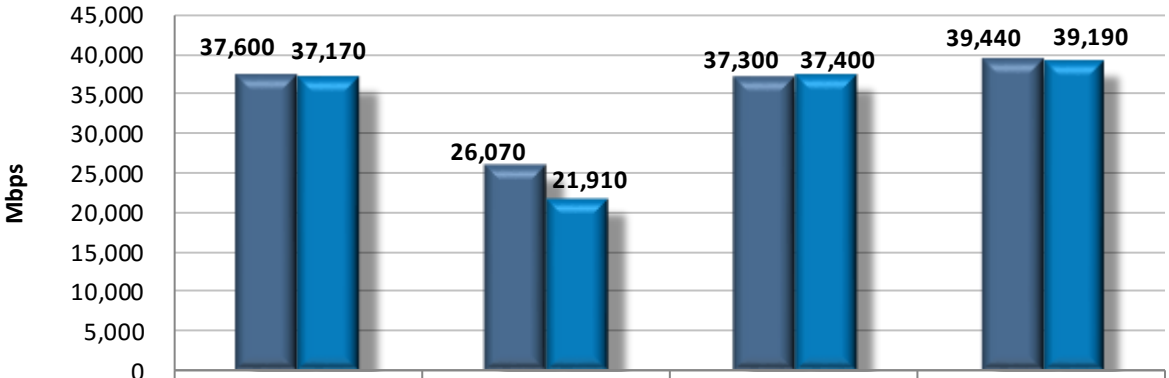


	HTTP 250 CPS	HTTP 500 CPS	HTTP 1000 CPS
IPv4 (CPS)	4,345	10,280	14,520
IPv6 (CPS)	4,239	9,878	13,620
IPv4 (Mbps)	17,380	20,560	14,520
IPv6 (Mbps)	16,956	19,756	13,620

Figure 13 – HTTP Capacity with HTTP Persistent Connections

Real-World Traffic Mixes

This test measures the performance of the device in a "real-world" environment by introducing additional protocols and real content, while still maintaining a precisely repeatable and consistent background traffic load. Different protocol mixes are utilized based on the intended location of the device to reflect real use cases. For details about real-world traffic protocol types and percentages, see the NSS Labs Data Center Security Gateway Test Methodology, available at www.nsslabs.com. Figure 14 depicts the results of the IPv4 and IPv6 tests for "real-world" traffic mixes.



	Real World Protocol Mix (Web Applications and Services)	Real World Protocol Mix (Financial)	Real World Protocol Mix (DCISP)	Real-World Protocol Mix (Streaming Media)
IPv4 (Mbps)	37,600	26,070	37,300	39,440
IPv6 (Mbps)	37,170	21,910	37,400	39,190

Figure 14 – "Real-World" Traffic Mixes

The FortiGate 3000D was tested by NSS and performed above the vendor-claimed IPv4 throughput for all "real-world" traffic mixes. The device performed above the IPv6 vendor-claimed throughput for the Web Applications and Services, DCISP, and Streaming Media "real-world" traffic mixes and below the IPv6 vendor-claimed throughput for the Financial mix.

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, it 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 DCSG devices must choose whether to risk denying legitimate traffic or risk allowing malicious traffic once they run low on resources. A DCSG 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 DCSG 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

This table 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.

The 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 3000D v5.4.5 GA Build 3273	8

Figure 16 – Sensor 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 3000D v5.4.5 GA Build 3273	$37,500	$15,703	$53,803	$15,703	$15,703	$85,209

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	Results
Firewall Policy Enforcement	
Baseline Policy	PASS
Simple Policy	PASS
Complex Policy	PASS
Static NAT	PASS
SYN Flood Protection	PASS
Address Spoofing Protection	PASS
NSS Exploit Block Rate	**97.97%**
False Positive Testing	PASS
Evasions and Attack Leakage	113/113
Resistance to Evasion	**PASS**
IP Packet Fragmentation	**PASS**
Ordered 8-byte fragments	PASS
Ordered 16-byte fragments	PASS
Ordered 24-byte fragments	PASS
Ordered 32-byte fragments	PASS
Out of order 8-byte fragments	PASS
Ordered 8-byte fragments, duplicate last packet	PASS
Out of order 8-byte fragments, duplicate last packet	PASS
Ordered 8-byte fragments, reorder fragments in reverse	PASS
Ordered 16-byte fragments, fragment overlap (favor new)	PASS
Ordered 16-byte fragments, fragment overlap (favor old)	PASS
Out of order 8-byte fragments, interleaved duplicate packets scheduled for later delivery	PASS
Ordered 8-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload.	PASS
Ordered 16-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload.	PASS
Ordered 24-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload.	PASS
Ordered 32-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload.	PASS
TCP Stream Segmentation	**PASS**
Ordered 1-byte segments, interleaved duplicate segments with invalid TCP checksums	PASS
Ordered 1-byte segments, interleaved duplicate segments with null TCP control flags	PASS
Ordered 1-byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream	PASS
Ordered 1-byte segments, duplicate last packet	PASS
Ordered 2-byte segments, segment overlap (favor new)	PASS
Ordered 1-byte segments, interleaved duplicate segments with out-of-window sequence numbers	PASS
Out of order 1-byte segments	PASS

Out of order 1-byte segments, interleaved duplicate segments with faked retransmits	PASS
Ordered 1-byte segments, segment overlap (favor new)	PASS
Out of order 1-byte segments, PAWS elimination (interleaved duplicate segments with older TCP timestamp options)	PASS
Ordered 16-byte segments, segment overlap (favor new (Unix))	PASS
Ordered 32-byte segments	PASS
Ordered 64-byte segments	PASS
Ordered 128-byte segments	PASS
Ordered 256-byte segments	PASS
Ordered 512-byte segments	PASS
Ordered 1024-byte segments	PASS
Ordered 2048-byte segments (sending MSRPC request with exploit)	PASS
Reverse Ordered 256-byte segments, segment overlap (favor new) with random data	PASS
Reverse Ordered 512-byte segments, segment overlap (favor new) with random data	PASS
Reverse Ordered 1024-byte segments, segment overlap (favor new) with random data	PASS
Reverse Ordered 2048-byte segments, segment overlap (favor new) with random data	PASS
Out of order 1024-byte segments, segment overlap (favor new) with random data, Initial TCP sequence number is set to 0xffffffff - 4294967295	PASS
Out of order 2048-byte segments, segment overlap (favor new) with random data, Initial TCP sequence number is set to 0xffffffff - 4294967295	PASS
RPC Fragmentation	**PASS**
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
MSRPC messages are sent in the big endian byte order, 16 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 32 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS

MSRPC messages are sent in the big endian byte order, 64 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 128 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 256 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 512 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 1024 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
URL Obfuscation	**PASS**
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 Evasion / Telnet Evasion	**PASS**
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
Layered Evasions	**PASS**
IP Fragmentation + TCP Segmentation	**PASS**
Ordered 8-byte fragments + Ordered TCP segments except that the last segment comes first	PASS

Ordered 24-byte fragments + Ordered TCP segments except that the last segment comes first	PASS
Ordered 32-byte fragments + Ordered TCP segments except that the last segment comes first	PASS
Ordered 8-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Reverse order TCP segments, segment overlap (favor new), Overlapping data is set to zero bytes	PASS
Ordered 16-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to zero bytes	PASS
Ordered 24-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to zero bytes	PASS
Ordered 32-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to zero bytes	PASS
Ordered 8-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random alphanumeric	PASS
Ordered 16-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random alphanumeric	PASS
Ordered 32-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random alphanumeric	PASS
Ordered 8-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random bytes	PASS
Ordered 16-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random bytes	PASS
Ordered 24-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random bytes	PASS
Ordered 32-byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random bytes	PASS
IP Fragmentation + MSRPC Fragmentation	**PASS**
Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 8 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 2048 bytes of payload.	PASS
Ordered 16 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 16 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 2048 bytes of payload.	PASS
Ordered 32 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 32 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 64 bytes of payload.	PASS
Ordered 64 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 64 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 64 bytes of payload.	PASS

Ordered 128 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + MSRPC messages are sent in the big endian byte order with 1024 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 128 bytes of payload.	PASS	
Ordered 256 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + MSRPC messages are sent in the big endian byte order with 1024 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 256 bytes of payload.	PASS	
Ordered 512 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + MSRPC messages are sent in the big endian byte order with 1024 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 512 bytes of payload.	PASS	
Ordered 1024 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + MSRPC messages are sent in the big endian byte order with 1024 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 1024 bytes of payload.	PASS	
IP Fragmentation + SMB Evasions	**PASS**	
Ordered 1024 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + SMB chaff message before real messages. The chaff is a WriteAndX message with a broken write mode flag, and has random MSRPC request-like payload	PASS	
Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + A chaffed NetBIOS message is sent before the first actual NetBIOS message. The chaff message is an unspecified NetBIOS message with MSRPC request like payload	PASS	
Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + A chaffed NetBIOS message is sent before the first actual NetBIOS message. The chaff message is an unspecified NetBIOS message with HTTP GET request like payload	PASS	
TCP Segmentation + SMB / NETBIOS Evasions	**PASS**	
Reverse Ordered 2048 byte TCP segments, segment overlap (favor new) with random data + A chaffed NetBIOS message is sent before the first actual NetBIOS message. The chaff message is an unspecified NetBIOS message with MSRPC request like payload	PASS	
IP Fragmentation + MSRPC Fragmentation	**PASS**	
Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 8 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 2048 bytes of payload.	PASS	

Performance	IPv4 Results	IPv6 Results
Raw Packet Processing Performance (UDP Traffic)	**Mbps**	
64-Byte Packets	35,900	Not Tested
128-Byte Packets	36,660	Not Tested
256-Byte Packets	37,300	Not Tested
512-Byte Packets	37,460	Not Tested
1024-Byte Packets	37,660	Not Tested
1514-Byte Packets	37,700	Not Tested
Latency – UDP	**Microseconds**	
64-Byte Packets	3.03	Not Tested

128-Byte Packets	3.17	Not Tested
256-Byte Packets	3.56	Not Tested
512-Byte Packets	4.23	Not Tested
1024-Byte Packets	5.52	Not Tested
1514-Byte Packets	6.73	Not Tested
Maximum Capacity	**CPS**	
Theoretical Max. Concurrent TCP Connections	16,604,512	16,992,013
Theoretical Max. Concurrent TCP Connections w/Data	21,514,647	22,015,692
Maximum TCP Connections per Second	216,000	216,000
Maximum HTTP Connections per Second	170,000	173,300
Maximum HTTP Transactions per Second	390,300	375,400
HTTP Capacity	**CPS**	
2,500 Connections per Second – 44 KB Response	43,600	45,110
5,000 Connections per Second – 21 KB Response	72,620	72,800
10,000 Connections per Second – 10 KB Response	100,400	100,800
20,000 Connections per Second – 4.5 KB Response	123,200	124,800
40,000 Connections per Second – 1.7 KB Response	147,200	148,900
Application Average Response Time – HTTP (at 90% Max Load)	**Milliseconds**	
2.500 Connections per Second – 44 KB Response	2.84	3.26
5,000 Connections per Second – 21 KB Response	3.15	3.22
10,000 Connections per Second – 10 KB Response	3.02	2.86
20,000 Connections per Second – 4.5 KB Response	2.51	2.38
40,000 Connections per Second – 1.7 KB Response	3.35	2.95
HTTP Capacity with HTTP Persistent Connections	**CPS**	
250 Connections per Second	4,345	4,239
500 Connections per Second	10,280	9,878
1000 Connections per Second	14,520	13,620
"Real-World" Traffic	**Mbps**	
Real-World Protocol Mix (Web Applications and Services)	37,600	37,170
Real-World Protocol Mix (Financial)	26,070	21,910
Real-World Protocol Mix (DCISP)	37,300	37,400
Real-World Protocol Mix (Streaming Media)	39,440	39,190
Stability and Reliability		
Behavior of the State Engine under Load		
Attack Detection/Blocking – Normal Load	PASS	
State Preservation – Normal Load	PASS	
Pass Legitimate Traffic – Normal Load	PASS	
State Preservation – Maximum Exceeded	PASS	
Drop Traffic – Maximum Exceeded	PASS	
Total Cost of Ownership		
Ease of Use		
Initial Setup (Hours)	8	
Time Required for Upkeep (Hours per Year)	Contact NSS	

Time Required to Tune (Hours per Year)	Contact NSS
Expected Costs	**US$**
Initial Purchase (hardware as tested)	$37,500
Installation Labor Cost (@$75/hr)	$600
Annual Cost of Maintenance and Support (hardware/software)	$8,203
Annual Cost of Updates (IPS/AV/etc.)	$7,500
Initial Purchase (enterprise management system)	See Comparative
Annual Cost of Maintenance and Support (enterprise management system)	See Comparative
Total Cost of Ownership	**US$**
Year 1	$53,803
Year 2	$15,703
Year 3	$15,703
3-Year Total Cost of Ownership	$85,209

Figure 18 – Detailed Scorecard

Test Methodology

Data Center Security Gateway (DCSG) Test Methodology v1.0

A copy of the test methodology is available on the NSS Labs website 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