NEXT GENERATION FIREWALL TEST REPORT

Fortinet FortiGate 500E V5.6.3GA build7858

JULY 17, 2018

Authors – Devon James, Michael Shirley, Tim Otto

Overview

NSS Labs performed an independent test of the Fortinet FortiGate 500E V5.6.3GA build7858. The product was subjected to thorough testing at the NSS facility in Austin, Texas, based on the Next Generation Firewall Test Methodology v8.0, the Secure Sockets Layer/Transport Layer Security Performance Test Methodology v1.3, and the Evasions Test Methodology v1.1, all of which are available at www.nsslabs.com. Testing 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. For details on performance with Secure Sockets Layer (SSL)/Transport Layer Security (TLS) encryption enabled, please see the individual SSL/TLS Test Reports.

NSS research indicates that next generation firewalls are typically deployed to protect users rather than data center assets and that the majority of enterprises will not separately tune intrusion prevention system (IPS) modules within their NGFWs. Therefore, during NSS testing, NGFW products are configured with the vendor's pre-defined or recommended (i.e., "out-of-the-box") settings in order to provide readers with relevant security effectiveness and performance dimensions based on their expected usage.

<table>
<tr><th>Product</th><th colspan="2">NSS-Tested Throughput</th><th>3-Year TCO (US$)</th></tr>
<tr><td rowspan="3">Fortinet FortiGate 500E
V5.6.3GA build7858</td><td colspan="2">6,753 Mbps</td><td>$11,364</td></tr>
<tr><td>Exploit Block Rate[1]</td><td>Evasions Blocked[2]</td><td>Stability & Reliability</td></tr>
<tr><td>99.31%</td><td>190/190</td><td>PASS</td></tr>
</table>

Figure 1 – Overall Test Results

Using the recommended policy, the FortiGate 500E blocked 99.31% of attacks. The device proved effective against 190 out of 190 evasions tested. The device passed all stability and reliability tests.

The FortiGate 500E is rated by NSS at 6,753 Mbps, which is higher than the vendor-claimed performance; Fortinet rates this device at 5,200Mbps. *NSS-Tested Throughput* is calculated as a weighted average of the traffic that NSS expects an NGFW to experience in an enterprise environment. For more details, please see Appendix A: Product Scorecard.

[1] Exploit block rate is defined as the total number of samples (live exploits and exploits from NSS Exploit Library) that are blocked under test.

[2] 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

The firewall market 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 they should therefore be viewed as an implementation of policy.

A firewall is a mechanism used to protect a trusted network from an untrusted network, while allowing authorized communications to pass from one side to the other, thus facilitating secure business use of the Internet. With the emergence of HTML 5, web browsers and security threats, however, firewalls are evolving further. NGFWs traditionally have been deployed to defend the network on the edge, but some enterprises have expanded their deployment to include internal segmentation.

As Web 3.0 trends push critical business applications through firewall ports that previously were reserved for a single function, such as HTTP, legacy firewall technology is effectively blinded. It is unable to differentiate between actual HTTP traffic and non-HTTP services tunneling over port 80, such as VoIP or instant messaging. Today, application-level monitoring must be performed in addition to analysis of port and destination. Firewalls are evolving to address this increased complexity.

It is no longer possible to rely on port and protocol combinations alone to define network applications. The NGFW must be capable of determining which applications are running regardless of which ports they are using and thus secure them effectively. This section verifies that the device is capable of enforcing the security policy effectively.

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 1,900 exploits, this is the industry's most comprehensive test to date.

Product	Total Number of Attacks Run	Total Number of Attacks Blocked	Block Percentage
Fortinet FortiGate 500E V5.6.3GA build7858	2,074	2,056	99.13%

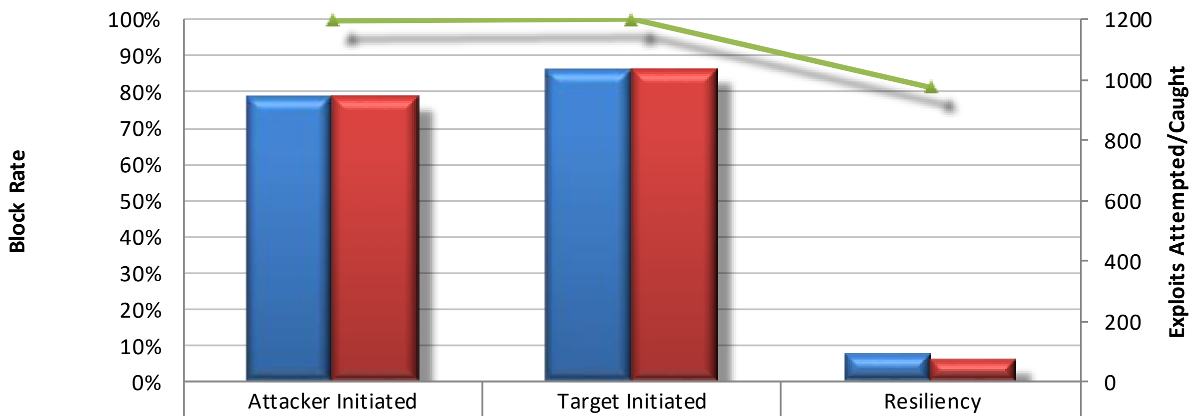
Figure 2 – Number of Attacks Blocked (%)

False Positive Testing

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

Coverage by Attack Vector and Resiliency

Because a failure to block attacks could result in significant compromise and could severely impact critical business systems, NGFWs should be evaluated against a broad set of exploits. Exploits can be categorized as either *attacker-initiated* or *target-initiated*. Attacker-initiated exploits are threats executed remotely against a vulnerable application and/or operating system by an individual, while target-initiated exploits are initiated by the vulnerable target. Target-initiated exploits are the most common type of attack experienced by the end user, and the attacker has little or no control as to when the threat is executed. 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.



	Attacker Initiated	Target Initiated	Resiliency
Attempted	947	1036	91
Caught	946	1036	74
Coverage	99.9%	100.0%	81.3%

Figure 3 – Coverage by Attack Vector and Resiliency

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 client or server. Slightly less serious are attacks that result in individual service compromise but not arbitrary system-level command execution, but this distinction is becoming less relevant in the modern threat landscape. 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.

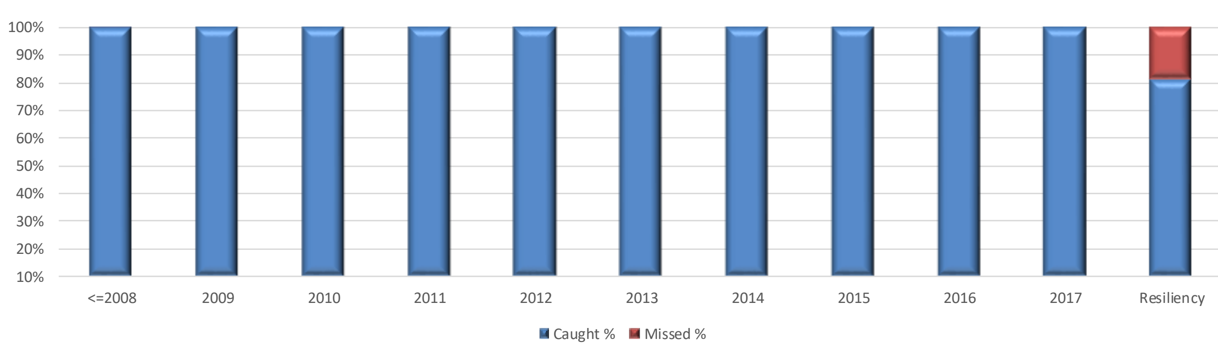


Figure 4 – Product Coverage by Date

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 500E for five of the top vendors targeted in this test. More than 70 vendors are represented in the test. Clients can contact NSS for more information.

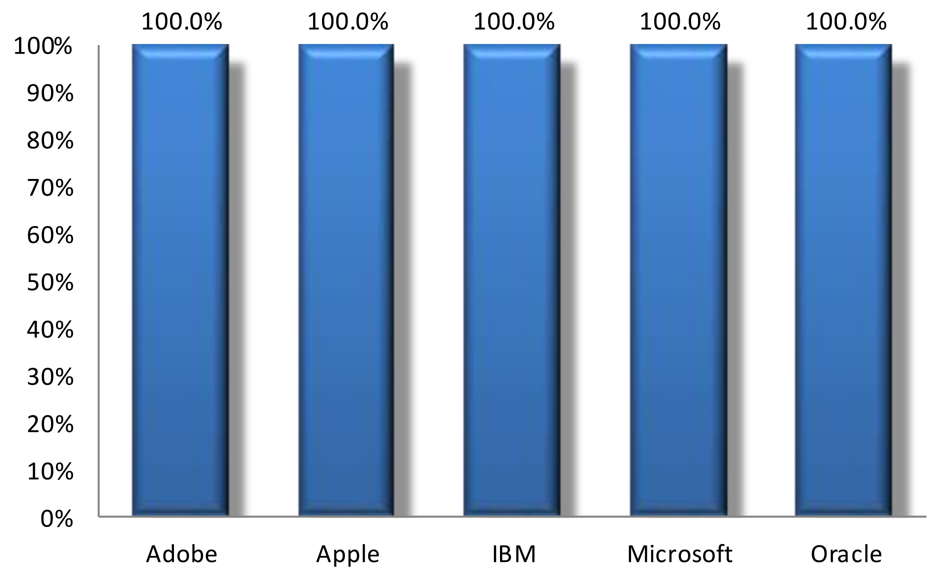


Figure 5 – Product Coverage by Target Vendor

Live Exploits

This test uses NSS' continuous live testing capabilities to determine how effective products are at blocking exploits that are being used, or that have been used in active attack campaigns.[3]

Protection from web-based exploits targeting client applications, also known as "drive-by" downloads, can be effectively measured in NSS' unique live test harness through a series of procedures that measure the stages of protection.

Unlike traditional malware that is downloaded and installed, "drive-by" attacks first exploit a vulnerable application then silently download and install malware. For more information, see the Comparative Report on Security.

Product	Block Percentage
Fortinet FortiGate 500E V5.6.3GA build7858	100.00%

Figure 6 – Number of Attacks Blocked (%)

[3] See the NSS Continuous Security Validation Platform for more details.

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 often 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 NGFW product category.

Providing exploit protection results without fully factoring in evasions can be misleading. The more classes of evasion that are missed (such as HTTP evasions, IP packet fragmentation, TCP stream segmentation, RPC fragmentation, URL obfuscation, HTML obfuscation, resiliency, 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 (HTTP 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.

TCP Split Handshake attacks can deceive the IPS engine into believing that the traffic flow is reversed and the IPS does not need to scan the content, which exposes the NGFW to previously known attacks.

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 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 7 provides the results of the evasion tests for the FortiGate 500E.

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

Figure 7 – 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 egress filtering and should ensure their configurations mitigate these risks.

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.

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 along with variable source and destination IP addresses 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 real-world network condition. The aim of the test is to determine the raw packet processing capability of each inline port pair of the device as well as the device's effectiveness at forwarding packets quickly, in order to provide the highest level of network performance with the least amount of latency.

	64 Byte Packets	128 Byte Packets	256 Byte Packets	512 Byte Packets	1024 Byte Packets	1514 Byte Packets
Mbps	20,000	20,000	20,000	20,000	20,000	20,000
Latency (µs)	7	7	7	8	9	10

Figure 8 – Raw Packet Processing Performance (UDP Traffic)

Raw Packet Processing Performance (UDP Latency)

NGFWs 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 9 depicts UDP latency (in microseconds) as recorded during the UDP throughput tests at 90% of maximum load.

Latency – UDP	Microseconds
64-Byte Packets	6.84
128-Byte Packets	6.88
256-Byte Packets	7.16
512-Byte Packets	7.54
1024-Byte Packets	8.92
1514-Byte Packets	10.04

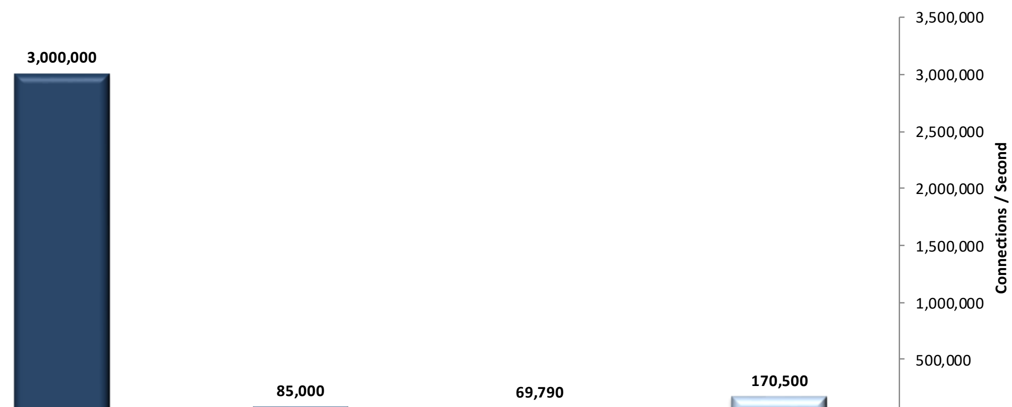
Figure 9 – UDP Latency in Microseconds

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



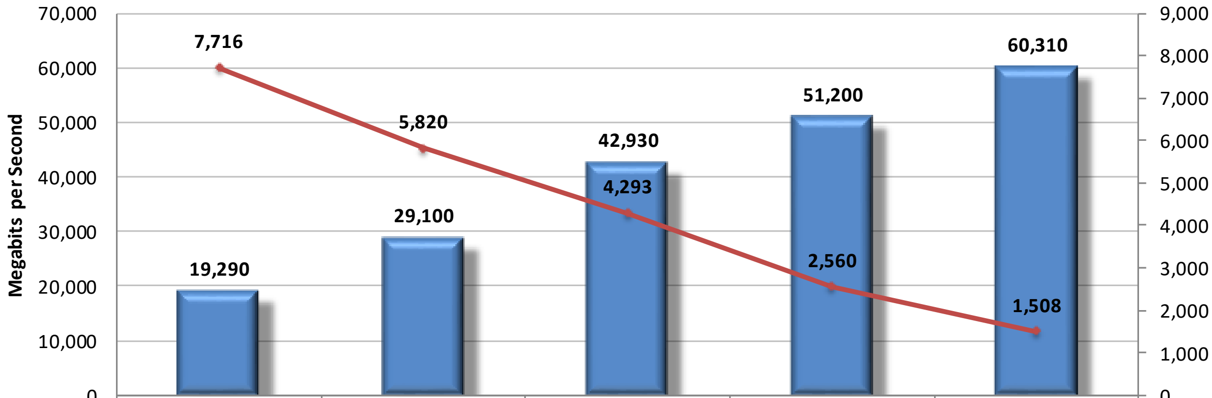
	Max Concurrent TCP Connections	Max TCP Connections per Second	Max HTTP Connections per Second	Max HTTP Transactions per Second
Without Data	3,000,000	85,000	69,790	170,500

Figure 10 – Concurrency and Connection Rates

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 multiple tests using genuine session-based traffic with varying session lengths, the device is forced to track valid HTTP sessions, thus ensuring a higher workload than for simple packet-based background traffic.

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.



	44 KB Response	21 KB Response	10 KB Response	4.5 KB Response	1.7 KB Response
CPS	19,290	29,100	42,930	51,200	60,310
Mbps	7,716	5,820	4,293	2,560	1,508

Figure 11 – HTTP Capacity

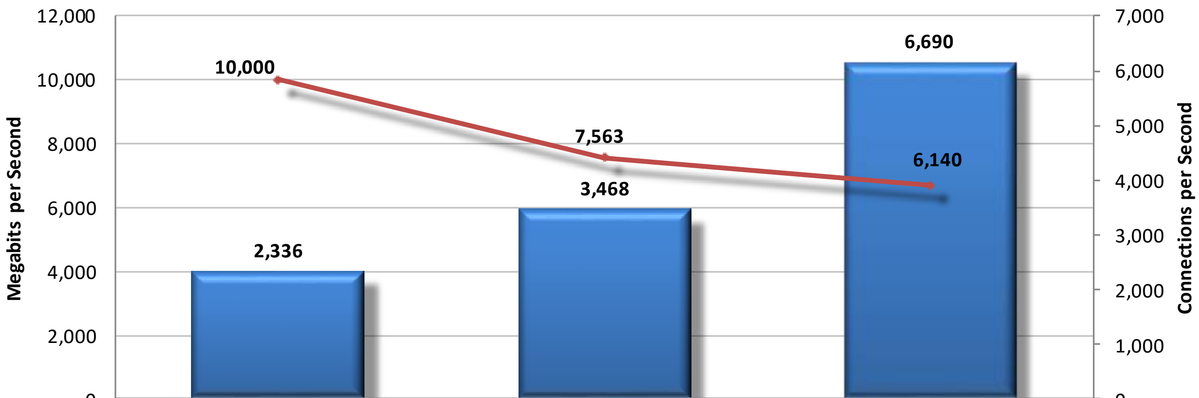
Application Average Response Time – HTTP

Application Average Response Time – HTTP (at 90% Maximum Load)	Milliseconds
2,500 Connections per Second – 44 KB Response	2.35
5,000 Connections per Second – 21 KB Response	1.54
10,000 Connections per Second – 10 KB Response	1.19
20,000 Connections per Second – 4.5 KB Response	0.79
40,000 Connections per Second – 1.7 KB Response	0.74

Figure 12 – Average Application Response Time (Milliseconds)

HTTP Capacity with HTTP Persistent Connections

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.

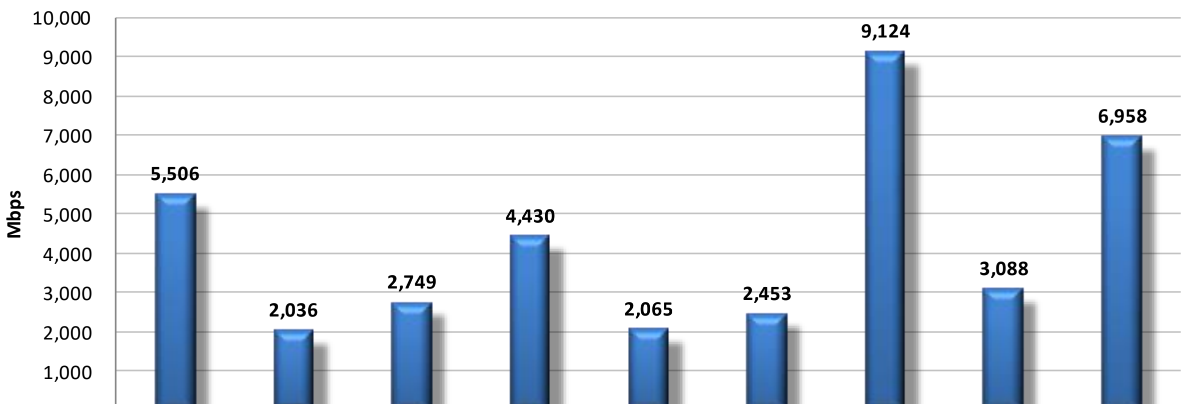


	HTTP 250 CPS	HTTP 500 CPS	HTTP 1000 CPS
CPS	2,336	3,468	6,140
Mbps	10,000	7,563	6,690

Figure 13 – HTTP Capacity 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 Next Generation Firewall Test Methodology, available at www.nsslabs.com.



	Telephony	Financial	Email	File Sharing	Fileserver	Remote Console	Video	Meetings	Database
Mbps	5,506	2,036	2,749	4,430	2,065	2,453	9,124	3,088	6,958

Figure 14 – Single Application Flows

SSL/TLS

Use of the Secure Sockets Layer (SSL) protocol and its newer iteration, Transport Layer Security (TLS), has risen in accordance with the increasing need for privacy online. Modern cybercampaigns frequently focus on attacking users through the most common web protocols and applications. NSS continues to receive inquiries from enterprise customers during their assessments of vendors that provide SSL/TLS decryption and protection technologies. Figure 15 provides the results of the SSL performance testing for the FortiGate 500E.

Product	NSS-Tested SSL/TLS Throughput (Mbps)
Fortinet FortiGate 500E V5.6.3GA build7858	5,773

Figure 15 – NSS-Tested SSL/TLS Throughput (Mbps)

The FortiGate 500E is rated by NSS at 5,773 Mbps with SSL/TLS enabled.

NSS-Tested SSL/TLS Throughput is calculated as a weighted average of the traffic that NSS expects an NGFW to experience in an enterprise environment. The device supports all SSL/TLS functionality. For further details on SSL performance, please see the SSL Performance Test Report for this device.

Stability and Reliability

Long-term stability is particularly important for an inline device, where failure can produce a network outage. These tests verify the device's ability to block malicious traffic while under extended load. Products that cannot sustain legitimate traffic while under test will fail.

The device is required to remain operational and stable throughout all these tests, and to block 100% of previously known malicious attacks, 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
Blocking under Extended Attack	PASS
Passing Legitimate Traffic under Extended Attack	PASS
Behavior of the State Engine under Load	
• Attack Detection/Blocking – Normal Load	PASS
• State Preservation – Normal Load	PASS
• Pass Legitimate Traffic – Normal Load	PASS
• Drop Traffic – Maximum Exceeded	PASS
Power Fail	PASS
Backup / Restore	PASS
Persistence of Data	PASS
Stability	PASS

Figure 16 – Stability and Reliability Results

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 17 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 500E V5.6.3GA build7858	8

Figure 17 – 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	Year 1 Cost	Year 2 Cost	Year 3 Cost	3-Year TCO
Fortinet FortiGate 500E V5.6.3GA build7858	$7,688	$1,838	$1,838	$11,364

Figure 18 –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.

For the FortiGate 500E, updates for the first year are included in the initial purchase price and are not counted again in *Year 1 Cost*.

- **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

Description	Result
Security Effectiveness	
False Positive Testing	PASS
Exploit Block Rate	**99.31%**
NSS Exploit Library Block Rate	99.13%
Live Exploits Block Rate	100.00%
Coverage by Attack Vector (NSS Exploit Library)	
Attacker-Initiated	99.89%
Target-Initiated	100.00%
Resiliency	81.32%
Combined Total	99.13%
Coverage by Impact Type	
System Exposure	Contact NSS
Service Exposure	Contact NSS
System or Service Fault	Contact NSS
Coverage by Date	Contact NSS
Coverage by Target Vendor	Contact NSS
Coverage by Result	Contact NSS
Coverage by Target Type	Contact NSS
Evasions and Attack Leakage	
Resistance to Evasions	**PASS**
IP Packet Fragmentation/ TCP Segmentation	**PASS**
(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; delay first fragment)	PASS
(overlapping small IP fragments favoring new data in reverse order; delay last fragment)	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
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
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 Evasions	**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
HTTP Evasions	**PASS**
(HTTP/0.9 response (no response headers))	PASS
(Declared HTTP/0.9 response; but includes response headers; space (hex '20') after server header)	PASS
(HTTP/1.1 chunked response with chunk sizes followed by a space (hex '20'))	PASS
(HTTP/1.1 chunked response with chunk sizes followed by a tab (hex '09'))	PASS
(HTTP/1.1 chunked response with chunk sizes followed by an 'x' (hex '78'))	PASS
(HTTP/1.1 chunked response with chunk sizes followed by a comma (hex '2c'))	PASS
(HTTP/1.1 chunked response with chunk sizes followed by null character (hex '00'))	PASS
(HTTP/1.1 chunked response with ‘Server’ header before Status-Line; with chunk sizes followed by a vertical tab (hex '0b'))	PASS
(HTTP/1.1 chunked response with chunk sizes followed by form feed (hex '0c'))	PASS
(HTTP/1.1 chunked response with final chunk size of '00' (hex '30 30' rather than hex '30'))	PASS
(HTTP/1.1 chunked response with final chunk size of '00000000000000000000' (rather than '0'))	PASS
(HTTP/1.1 chunked response with chunk sizes followed by a space (hex '20') then an 'x' (hex '78'))	PASS
(HTTP/1.1 response with line folded transfer-encoding header declaring chunking ('Transfer-Encoding: ' followed by CRLF (hex '0d 0a') followed by space (hex '20') followed by 'chunked' followed by CRLF (hex '0d 0a')); served without chunking)	PASS
(HTTP/1.1 response with transfer-encoding header declaring chunking with lots of whitespace ('Transfer-Encoding: ' followed by 500 spaces (hex '20' * 500) followed by 'chunked' followed by CRLF (hex '0d 0a')); served chunked)	PASS
(HTTP/1.0 response declaring chunking; served without chunking)	PASS
(HTTP/1.0 response declaring chunking with content-length header; served without chunking)	PASS
(<tab>Transfer-Encoding: chunked as first header line; served chunked)	PASS
(<tab>Transfer-Encoding: chunked as continuation of some header line; served chunked)	PASS
(line with empty field name (single colon on line); followed by TE chunked; served chunked)	PASS
(TE chunked prefixed with <CR><CR>;served chunked)	PASS
(HTTP/1.1\nTransfer-Encoding:chunked; served chunked)	PASS
(HTTP/1.1 200 OK\r\nTransfer-Encoding:chunked; served chunked)	PASS
(single \n instead of \r\n and chunked)	PASS
(HTTP/1.1\rTransfer-Encoding: chunked; served chunked)	PASS
(double <LF> before header; chunked)	PASS

(double <CR><LF> before header; chunked)	PASS
(junk followed by single <CR><LF> before header; chunked)	PASS
(SIP/2.0 200 ok followed by single <CR><LF> before header; chunked)	PASS
(space+junk followed by single <CR><LF> before header; chunked)	PASS
(space+"SIP/2.0 200 ok" followed by single <CR><LF> before header; chunked)	PASS
(single <LF> before header; chunked)	PASS
(H before header; chunked)	PASS
(HT before header; chunked)	PASS
(HTT before header; chunked)	PASS
(HTTX before header; chunked)	PASS
(HTTXY before header; chunked)	PASS
(HTTP/1.1 response with content-encoding header for gzip; followed by content-encoding header for deflate; no space between ':' and declaration of encoding types; served with no compression)	PASS
(HTTP/1.1 response with content-encoding declaration of "gzip x"; served uncompressed)	PASS
(header end \n\r\n; gzip)	PASS
(header end \n\r\n; gzip with content-length)	PASS
(header end \n\013\n\n and gzip)	PASS
(header end \n\013\n\n and gzip with content length)	PASS
(header end \r\n\013\r\n\r\n and gzip)	PASS
(header end \r\n\013\r\n\r\n and gzip with content-length)	PASS
(header end \n\r\r\n; gzip)	PASS
(header end \n\r\r\n; gzip with content-length)	PASS
(header end "\n\x20 \n" and gzip)	PASS
(header end "\n\x20\n" and gzip with content-length)	PASS
(header end \n\011\n and gzip)	PASS
(header end \n\011\n and gzip with content-length)	PASS
(header end \n\n; gzip)	PASS
(HTTP/1.0 response with status code 100 followed by message-body; no content-length header)	PASS
(HTTP/1.0 response with status code 206 followed by message-body; no content-length header)	PASS
(HTTP/1.0 response with status code 304 followed by message-body; no content-length header)	PASS
(HTTP/1.0 response with status code 404 followed by message-body; no content-length header)	PASS
(HTTP/1.0 response with status code 500 followed by message-body; no content-length header)	PASS
(HTTP/1.1 response with status code 600 followed by a space; followed by message-body)	PASS
(HTTP/1.1 response with status code 900 followed by a space; followed by message-body)	PASS
(status code 101 with body)	PASS
(status code 102 with body)	PASS
(HTTP/1.1 response with content-length header size declaration followed by space and letter A (hex '20 41'))	PASS
(Chunked Header and HTTP/1.01. Served chunked)	PASS
(Chunked Header and HTTP/1.10. Served chunked)	PASS
(Chunked Header and HTTP/01.1. Served chunked and with gzip)	PASS
(Chunked Header and HTTP/11.01. Served chunked and with gzip)	PASS
(Chunked Header and HTTP/9.9. Served chunked and with gzip)	PASS

(version HTTP/1.010 instead of HTTP/1.1 and chunked)	PASS
(version HTTP/2.B instead of HTTP/1.1 and chunked)	PASS
(version HTTP/9.-1 instead of HTTP/1.1 and chunked)	PASS
(double Transfer-Encoding: first empty; last chunked. Served with content-length and gzipped; not chunked)	PASS
HTML Evasions	**PASS**
(UTF-8 encoding)	PASS
(UTF-8 encoding with BOM)	PASS
(UTF-16 encoding with BOM)	PASS
(UTF-8 encoding; no http or html declarations)	PASS
(UTF-8 encoding with BOM; no http or html declarations)	PASS
(UTF-16 encoding with BOM; no http or html declarations)	PASS
(padded with 1MB)	PASS
(padded with 15MB)	PASS
(padded with 30MB)	PASS
(padded with 1MB and chunked)	PASS
(padded with 15MB and chunked)	PASS
(padded with 30MB and chunked)	PASS
(padded with 1MB and compressed with gzip)	PASS
(padded with 15MB and compressed with gzip)	PASS
(padded with 30MB and compressed with gzip)	PASS
(padded with 1MB and compressed with deflate)	PASS
(padded with 15MB and compressed with deflate)	PASS
(padded with 30MB and compressed with deflate)	PASS
TCP Split Handshake	**PASS**
Resiliency	**FAIL**
Attacks on nonstandard ports	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	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	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	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	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	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		FAIL
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
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		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		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		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		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
Performance		
Raw Packet Processing Performance (UDP Traffic)	Weighting for NSS-Rated Throughput	Mbps
64-Byte Packets	0%	20,000
128-Byte Packets	1%	20,000
256-Byte Packets	1%	20,000
512-Byte Packets	1%	20,000
1024-Byte Packets	3%	20,000
1514-Byte Packets	3%	20,000

Latency – UDP		Microseconds
64-Byte Packets		6.84
128-Byte Packets		6.88
256-Byte Packets		7.16
512-Byte Packets		7.54
1024-Byte Packets		8.92
1514-Byte Packets		10.04
Maximum Capacity		CPS
Theoretical Max. Concurrent TCP Connections		3,000,000
Maximum TCP Connections per Second		85,000
Maximum HTTP Connections per Second		69,790
Maximum HTTP Transactions per Second		170,500
HTTP Capacity	Weighting for NSS-Rated Throughput	CPS
2,500 Connections per Second – 44 KB Response	8%	19,290
5,000 Connections per Second – 21 KB Response	8%	29,100
10,000 Connections per Second – 10 KB Response	7%	42,930
20,000 Connections per Second – 4.5 KB Response	7%	51,200
40,000 Connections per Second – 1.7 KB Response	4%	60,310
Application Average Response Time – HTTP (at 90% Max Load)		Milliseconds
2.500 Connections per Second – 44 KB Response		2.35
5,000 Connections per Second – 21 KB Response		1.54
10,000 Connections per Second – 10 KB Response		1.19
20,000 Connections per Second – 4.5 KB Response		0.79
40,000 Connections per Second – 1.7 KB Response		0.74
HTTP Capacity with HTTP Persistent Connections		CPS
250 Connections per Second		2,336
500 Connections per Second		3,468
1000 Connections per Second		6,140
Single Application Flows	Weighting for NSS-Rated Throughput	Mbps
Telephony	17%	5,506
Financial	0%	2,036
Email	12%	2,749
File Sharing	7%	4,430
Fileserver	0%	2,065
Remote Console	1%	2,453
Video	16%	9,124
Meetings	1%	3,088
Database	3%	6,958

Stability and Reliability	
Blocking under Extended Attack	PASS
Passing Legitimate Traffic under Extended Attack	PASS
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
Power Fail	PASS
Backup/Restore	PASS
Persistence of Data	PASS
Stability	PASS
Total Cost of Ownership	
Ease of Use	
Initial Setup (Hours)	8
Expected Costs	
Initial Purchase (hardware as tested)	$5,250
Installation Labor Cost (@$75/hr)	$600
Annual Cost of Maintenance and Support (hardware/software)	$1,838
Annual Cost of Updates (IPS/AV/etc.)	$0
Total Cost of Ownership	
Year 1	$7,688
Year 2	$1,838
Year 3	$1,838
3-Year Total Cost of Ownership	$11,364

Figure 19 – Detailed Scorecard

Test Methodology

NSS Labs Next Generation Firewall (NGFW) Test Methodology v8.0

NSS Labs SSL/TLS Performance Test Methodology v1.3

NSS Labs Evasions Test Methodology v1.1

Contact Information

3711 South Mopac Expressway
Building 1, Suite 400
Austin, TX 78746
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.