NEXT GENERATION FIREWALL TEST REPORT

Sophos XG-750 Firewall v16.01

JUNE 6, 2017

Authors –Jeff Bowermon, Devon James, Ty Smith

Overview

NSS Labs performed an independent test of the Sophos XG-750 Firewall v16.01. The product was subjected to thorough testing at the NSS facility in Austin, Texas, based on the Next Generation Firewall (NGFW) Test Methodology v7.0, which is available at www.nsslabs.com. This test was conducted free of charge and NSS did not receive any compensation in return for Sophos' 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 NGFW devices 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.

Product	Exploit Block Rate[1]	NSS-Tested Throughput	3-Year TCO (US$)
Sophos XG-750 Firewall v16.01	96.19%	8,628 Mbps	$45,244

Firewall Policy Enforcement	Application Control	Evasions Blocked	Stability and Reliability
PASS	PASS	135/137[2]	PASS

Figure 1 – Overall Test Results

Using the recommended policy, the XG-750 Firewall blocked 96.19% of attacks. The device failed to protect against the HTML obfuscation evasion technique. The device passed all stability and reliability tests.

The XG-750 Firewall is rated by NSS at 8,628 Mbps, which is lower than the vendor-claimed performance; Sophos rates this device at 11.8 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 live exploits (CAWS) and exploits from the *NSS Exploit Library* 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 untested evasion techniques were applied during testing until 90 days after the publication of this document.

Table of Contents

Overview 2

Security Effectiveness 5

Firewall Policy Enforcement 5

Application Control 6

CAWS (Live Exploits) 6

NSS Exploit Library 7

False Positive Testing 7

Coverage by Attack Vector 8

Coverage by Impact Type 8

Coverage by Date 9

Coverage by Target Vendor 9

Resistance to Evasion Techniques 10

Performance 11

Raw Packet Processing Performance (UDP Throughput) 11

Raw Packet Processing Performance (UDP Latency) 12

Maximum Capacity 12

HTTP Capacity 14

Application Average Response Time – HTTP 14

HTTP Capacity with HTTP Persistent Connections 15

HTTPS Capacity with HTTPS Persistent Connections 15

Real-World Traffic Mixes 16

Stability and Reliability 17

Total Cost of Ownership (TCO) 18

Installation Hours 18

Total Cost of Ownership 19

Appendix A: Product Scorecard 20

Test Methodology 26

Contact Information 26

Table of Figures

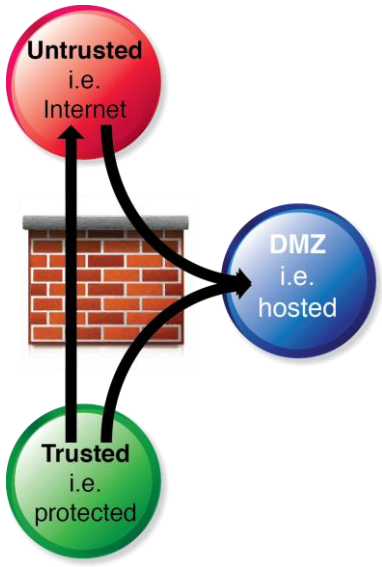
Figure 1 – Overall Test Results....2
Figure 2 – Firewall Policy Enforcement....5
Figure 3 – Application Control....6
Figure 4 – Number of Threat Encounters Blocked (%)....6
Figure 5 – Number of Exploits Blocked (%)....7
Figure 6 – Coverage by Attack Vector....8
Figure 7 – Product Coverage by Date....9
Figure 8 – Product Coverage by Target Vendor....9
Figure 9 – Resistance to Evasion Results....10
Figure 10 – Raw Packet Processing Performance (UDP Traffic)....11
Figure 11 – UDP Latency in Microseconds....12
Figure 12 – Concurrency and Connection Rates....13
Figure 13 – HTTP Capacity....14
Figure 14 – Average Application Response Time (Milliseconds)....14
Figure 15 – HTTP Capacity with HTTP Persistent Connections....15
Figure 16 – HTTPS Capacity with HTTPS Persistent Connections....15
Figure 17 – "Real-World" Traffic Mixes....16
Figure 18 – Stability and Reliability Results....17
Figure 19 – Sensor Installation Time (Hours)....18
Figure 20 –3-Year TCO (US$)....19
Figure 21 – Detailed Scorecard....25

Security Effectiveness

This section verifies that the device is capable of enforcing the security policy effectively.

Firewall Policy Enforcement

Policies are rules that are configured on a firewall to permit or deny access from one network resource to another, based on identifying criteria such as source, destination, and service. A term typically used to define the demarcation point of a network where policy is applied is *demilitarized zone* (DMZ). Policies are typically written to permit or deny network traffic from one or more of the following zones:



- **Untrusted –** This is typically an external network and is considered to be unknown and not secure. An example of an untrusted network would be the Internet.
- **DMZ –** This is a network that is being isolated by the firewall, restricting network traffic to and from hosts contained within the isolated network.
- **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 DMZ
- Trusted to DMZ

Note: Firewalls must provide at least one DMZ interface in order to provide a DMZ or "transition point" between untrusted and trusted networks.

Test Procedure	Result
Baseline Policy	PASS
Simple Policy	PASS
Complex Policy	PASS
Static NAT	PASS
Dynamic/Hide NAT	PASS
SYN Flood Protection	PASS
IP Address Spoofing Protection	PASS
TCP Split Handshake Spoof	PASS

Figure 2 – Firewall Policy Enforcement

Application Control

An NGFW must provide granular control based on applications as well as ports. This capability is needed to re-establish a secure perimeter where unwanted applications are unable to tunnel over HTTP/S. As such, granular application control is a requirement of an NGFW, as it enables the administrator to define security policies based on both applications and ports.

Test Procedure	Result
Block Unwanted Applications	PASS
Block Specific Actions	PASS

Figure 3 – Application Control

Our testing found that the XG-750 Firewall correctly enforced complex outbound and inbound policies consisting of multiple rules, objects, and applications. NSS engineers verified that the device successfully determined the correct application and took the appropriate action based on the policy.

CAWS (Live Exploits)

This test uses NSS' Cyber Advanced Warning System (CAWS) to determine how effectively products are able to block exploits that are being 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 – CAWS (Live Exploits).

Product	CAWS (Live Exploits) Threat Encounters	Total Number Blocked	Block Percentage
Sophos XG-750 Firewall v16.01	4534	4435	97.82%

Figure 4 – Number of Threat Encounters Blocked (%)

[3] See the NSS Cyber Advanced Warning System™ for more details.

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[4] as appropriate. With 2,097 exploits, this is the industry's most comprehensive test to date. Most notably, all of 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
Sophos XG-750 Firewall v16.01	2,097	1,983	94.56%

Figure 5 – Number of Exploits Blocked (%)

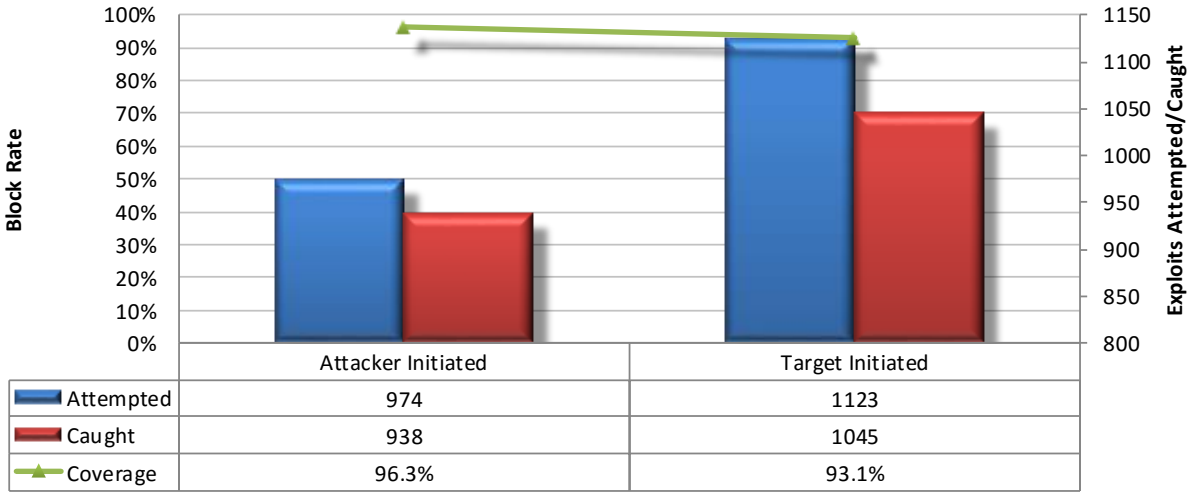
False Positive Testing

The XG-750 Firewall correctly identified traffic and did not fire alerts for non-malicious content.

[4] See the NSS Cyber Advanced Warning System™ for more details.

Coverage by Attack Vector

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.



	Attacker Initiated	Target Initiated
Attempted	974	1123
Caught	938	1045
Coverage	96.3%	93.1%

Figure 6 – Coverage by Attack Vector

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 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 7 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.

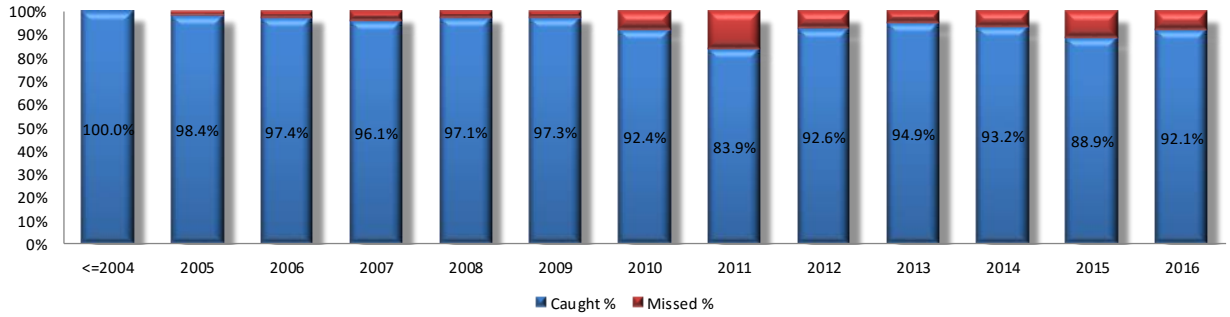


Figure 7 – Product Coverage by Date

Coverage by Target Vendor

Exploits within the *NSS Exploit Library* target a wide range of protocols and applications. Figure 8 depicts the coverage offered by the XG-750 Firewall 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 about this test.

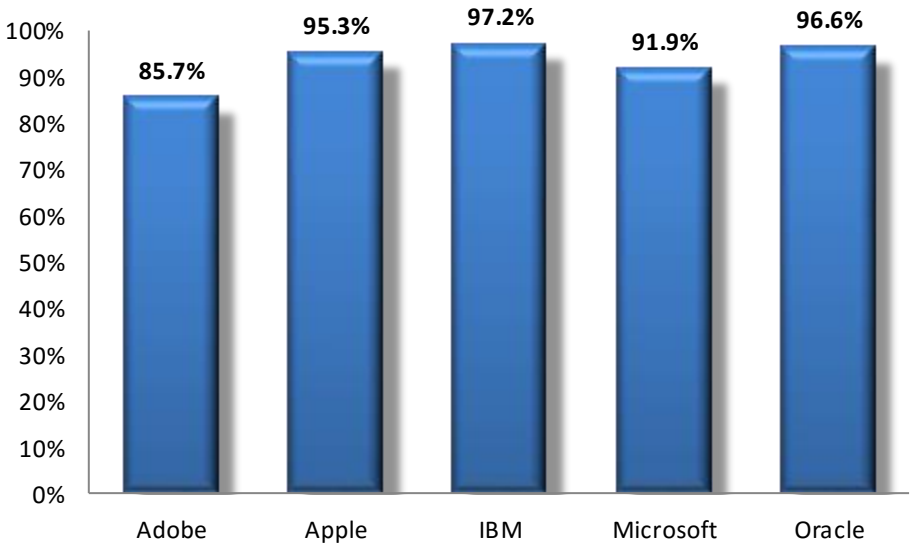


Figure 8 – Product Coverage by Target Vendor

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 NGFW product category.

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

Figure 9 provides the results of the evasion tests for the XG-750 Firewall. The XG-750 Firewall failed to block two of the 137 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
HTML Obfuscation	FAIL
HTTP Compression	PASS
FTP/Telnet Evasion	PASS
Payload Padding	PASS
IP Packet Fragmentation + TCP Segmentation	PASS
HTTP Evasion	PASS

Figure 9 – Resistance to Evasion Results

Performance

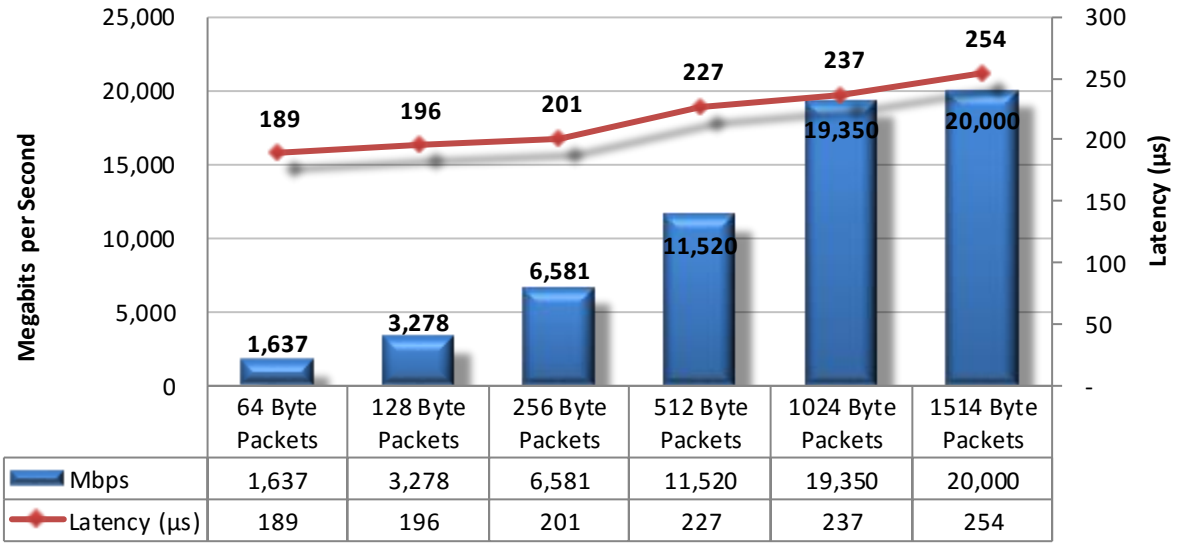
There is frequently a trade-off between security effectiveness and performance. Because of this trade-off, it is important to judge a product's security effectiveness within the context of its performance and vice versa. This ensures that new security protections do not adversely impact performance and that security shortcuts are not taken to maintain or improve performance.

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 state engine to do. 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.



	64 Byte Packets	128 Byte Packets	256 Byte Packets	512 Byte Packets	1024 Byte Packets	1514 Byte Packets
Mbps	1,637	3,278	6,581	11,520	19,350	20,000
Latency (µs)	189	196	201	227	237	254

Figure 10 – 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 11 depicts UDP latency (in microseconds) as recorded during the UDP throughput tests at 90% of maximum load.

Latency – UDP	Microseconds
64-Byte Packets	189.00
128-Byte Packets	196.00
256-Byte Packets	201.00
512-Byte Packets	227.00
1024-Byte Packets	237.00
1514-Byte Packets	254.00

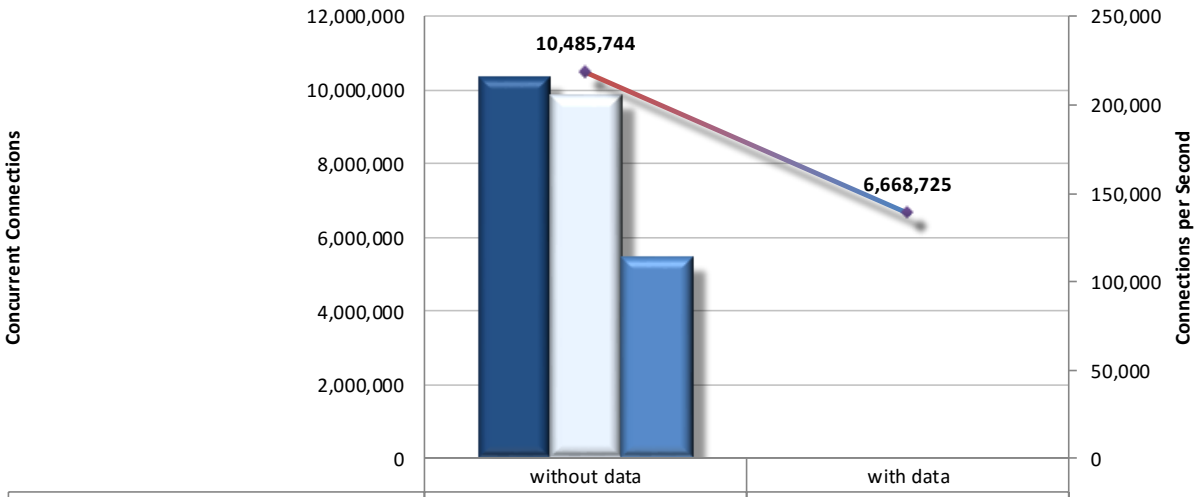
Figure 11 – UDP Latency in Microseconds

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.



	without data	with data
Maximum TCP Connections per Second	215,100	
Maximum HTTP Transactions per Second	204,900	
Maximum HTTP Connections per Second	113,000	
Max. Concurrent TCP Connections	10,485,744	6,668,725

Figure 12 – 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 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.

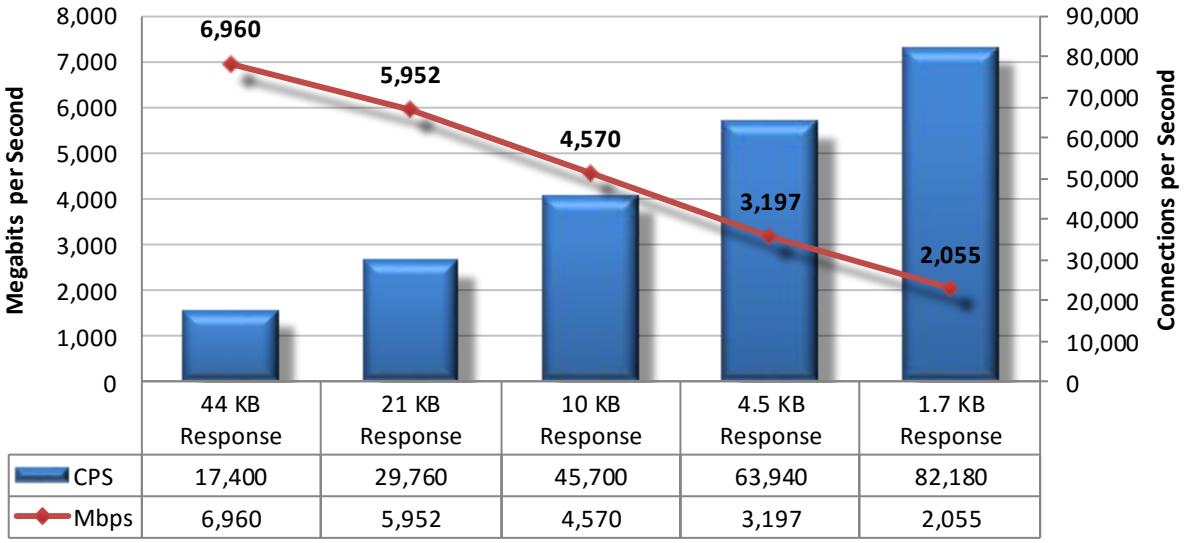


Figure 13 – HTTP Capacity

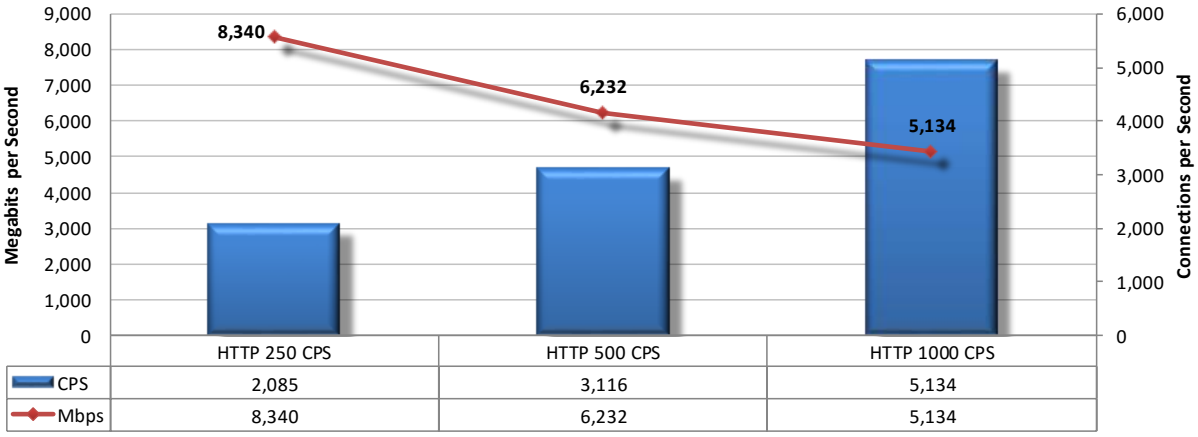
Application Average Response Time – HTTP

Application Average Response Time – HTTP (at 90% Maximum Load)	Milliseconds
2,500 Connections per Second – 44 KB Response	9.69
5,000 Connections per Second – 21 KB Response	7.45
10,000 Connections per Second – 10 KB Response	5.22
20,000 Connections per Second – 4.5 KB Response	3.33
40,000 Connections per Second – 1.7 KB Response	3.12

Figure 14 – 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.

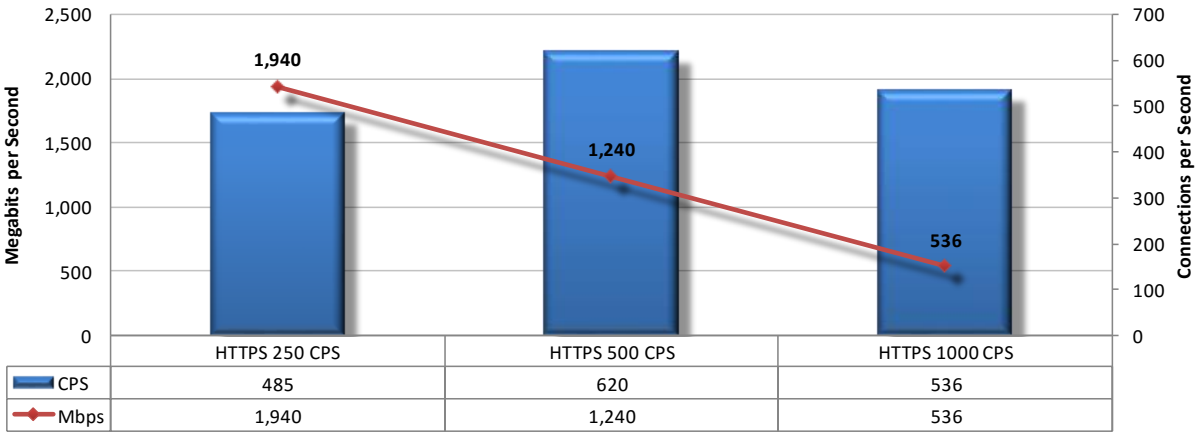


	HTTP 250 CPS	HTTP 500 CPS	HTTP 1000 CPS
CPS	2,085	3,116	5,134
Mbps	8,340	6,232	5,134

Figure 15 – HTTP Capacity with HTTP Persistent Connections

HTTPS Capacity with HTTPS Persistent Connections

This test will use HTTPS persistent connections, with each TCP connection containing 10 HTTPS GETs and associated responses.



	HTTPS 250 CPS	HTTPS 500 CPS	HTTPS 1000 CPS
CPS	485	620	536
Mbps	1,940	1,240	536

Figure 16 – HTTPS Capacity with HTTPS 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 (network core or perimeter) to reflect real use cases. For details about real-world traffic protocol types and percentages, see the NSS Labs Next Generation Firewall Test Methodology, available at www.nsslabs.com.

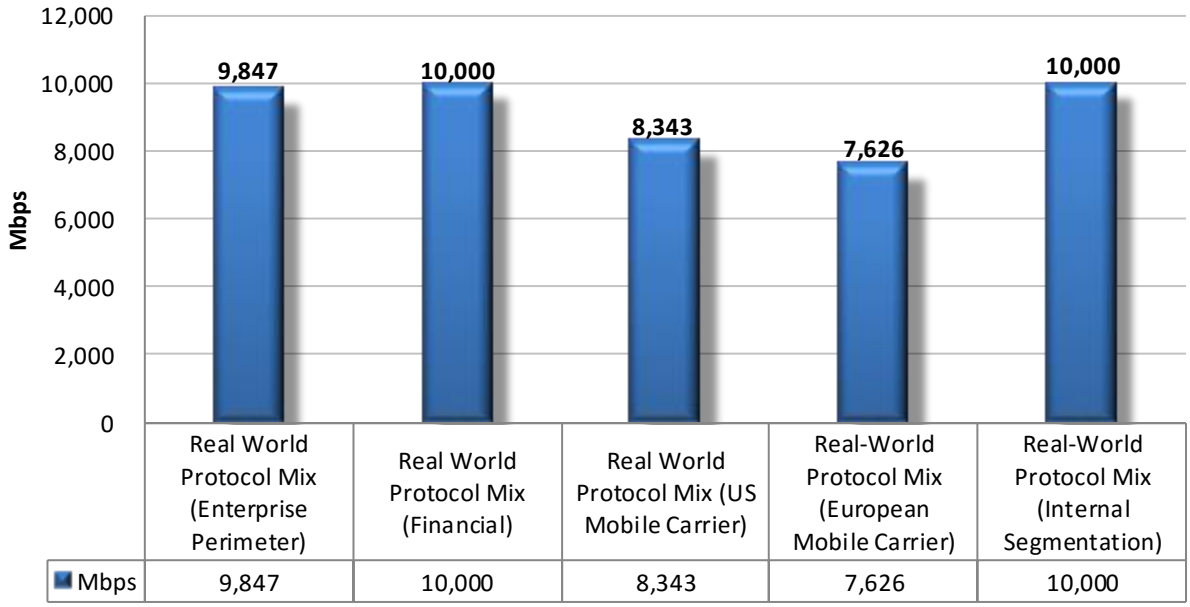


Figure 17 – "Real-World" Traffic Mixes

The XG-750 Firewall was tested by NSS and performed below the throughput claimed by the vendor for all "real-world" traffic mixes.

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.

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
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
Protocol Fuzzing and Mutation	PASS
Power Fail	PASS
Persistence of Data	PASS

Figure 18 – Stability and Reliability Results

These tests also determine the behavior of the state engine under load. All NGFW devices must choose whether to risk denying legitimate traffic or risk allowing malicious traffic once they run low on resources. An NGFW 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 NGFW 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)
Sophos XG-750 Firewall v16.01	8

Figure 19 – 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
Sophos XG-750 Firewall v16.01	$22,177	$7,489	$30,266	$7,489	$7,489	$45,244

Figure 20 –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

Description	Result
Security Effectiveness	
Firewall Policy Enforcement	**PASS**
Baseline Policy	PASS
Simple Policy	PASS
Complex Policy	PASS
Static NAT	PASS
Dynamic / Hide NAT	PASS
SYN Flood Protection	PASS
Address Spoofing Protection	PASS
TCP Split Handshake	PASS
Application Control	**PASS**
Block Unwanted Applications	PASS
Block Specific Action	PASS
Intrusion Prevention	
False Positive Testing	PASS
Exploit Block Rate	96.19%
CAWS (Live Exploits) Block Rate	97.82%
NSS Exploit Library Block Rate	94.56%
Coverage by Attack Vector (NSS Exploit Library)	
Attacker-Initiated	96.30%
Target-Initiated	93.05%
Combined Total	94.56%
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 Evasion	**FAIL**
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
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
HTML Obfuscation	**FAIL**
UTF-16 character set encoding (big-endian)	PASS
UTF-16 character set encoding (little-endian)	PASS
UTF-32 character set encoding (big-endian)	FAIL
UTF-32 character set encoding (little-endian)	FAIL
UTF-7 character set encoding	PASS
Chunked encoding (random chunk size)	PASS
Chunked encoding (fixed chunk size)	PASS
Chunked encoding (chaffing)	PASS
Compression (Deflate)	PASS
Compression (Gzip)	PASS
Base-64 Encoding	PASS
Base-64 Encoding (shifting 1 bit)	PASS
Base-64 Encoding (shifting 2 bits)	PASS
Base-64 Encoding (chaffing)	PASS
Combination UTF-7 + Gzip	PASS
HTTP Compression	**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
Payload Padding	**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
HTTP Evasion	**PASS**
Test Case 1	PASS
Test Case 2	PASS
Test Case 3	PASS
Test Case 4	PASS
Test Case 5	PASS
Test Case 6	PASS
Test Case 7	PASS
Test Case 8	PASS
Test Case 9	PASS
Test Case 10	PASS
Test Case 11	PASS
Test Case 12	PASS
Test Case 13	PASS
Test Case 14	PASS
Test Case 15	PASS
Test Case 16	PASS
Test Case 17	PASS
Test Case 18	PASS
Test Case 19	PASS
Test Case 20	PASS
Test Case 21	PASS
Test Case 22	PASS
Test Case 23	PASS
Test Case 24	PASS
Test Case 25	PASS
Test Case 26	PASS
Performance	
Raw Packet Processing Performance (UDP Traffic)	**Mbps**

64-Byte Packets	1,637
128-Byte Packets	3,278
256-Byte Packets	6,581
512-Byte Packets	11,520
1024-Byte Packets	19,350
1514-Byte Packets	20,000
Latency – UDP	**Microseconds**
64-Byte Packets	189.00
128-Byte Packets	196.00
256-Byte Packets	201.00
512-Byte Packets	227.00
1024-Byte Packets	237.00
1514-Byte Packets	254.00
Maximum Capacity	**CPS**
Theoretical Max. Concurrent TCP Connections	10,485,744
Theoretical Max. Concurrent TCP Connections w/Data	6,668,725
Maximum TCP Connections per Second	215,100
Maximum HTTP Connections per Second	113,000
Maximum HTTP Transactions per Second	204,900
HTTP Capacity	**CPS**
2,500 Connections per Second – 44 KB Response	17,400
5,000 Connections per Second – 21 KB Response	29,760
10,000 Connections per Second – 10 KB Response	45,700
20,000 Connections per Second – 4.5 KB Response	63,940
40,000 Connections per Second – 1.7 KB Response	82,180
Application Average Response Time – HTTP (at 90% Max Load)	**Milliseconds**
2.500 Connections per Second – 44 KB Response	9.69
5,000 Connections per Second – 21 KB Response	7.45
10,000 Connections per Second – 10 KB Response	5.22
20,000 Connections per Second – 4.5 KB Response	3.33
40,000 Connections per Second – 1.7 KB Response	3.12
HTTP Capacity with HTTP Persistent Connections	**CPS**
250 Connections per Second	2,085
500 Connections per Second	3,116
1000 Connections per Second	5,134
HTTPS Capacity with HTTPS Persistent Connections	**CPS**
250 Connections per Second	485
500 Connections per Second	620
1000 Connections per Second	536
“Real-World” Traffic	**Mbps**
“Real-World” Protocol Mix (Enterprise Perimeter)	9,847
“Real-World” Protocol Mix (Financial)	10,000
“Real-World” Protocol Mix (US Mobile Carrier)	8,343
“Real-World” Protocol Mix (EU Mobile Carrier)	7,626
“Real-World” Internal Segmentation Mix	10,000
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
Protocol Fuzzing and Mutation	PASS
Power Fail	PASS
Persistence of Data	PASS
Total Cost of Ownership	
Ease of Use	
Initial Setup (Hours)	8
Time Required for Upkeep (Hours per Year)	See Comparative
Time Required to Tune (Hours per Year)	See Comparative
Expected Costs	
Initial Purchase (hardware as tested)	$22,177
Installation Labor Cost (@$75/hr)	$600
Annual Cost of Maintenance and Support (hardware/software)	$7,489
Annual Cost of Updates (IPS/AV/etc.)	$0
Initial Purchase (enterprise management system)	See Comparative
Annual Cost of Maintenance and Support (enterprise management system)	See Comparative
Total Cost of Ownership	
Year 1	$30,266
Year 2	$7,489
Year 3	$7,489
3-Year Total Cost of Ownership	$45,244

Figure 21 – Detailed Scorecard

Test Methodology

Next Generation Firewall (NGFW) Test Methodology v7.0

A copy of the test methodology is available on the NSS Labs website at www.nsslabs.com.

Contact Information

NSS Labs, Inc.
206 Wild Basin Road
Building A, Suite 200
Austin, TX 78746 USA
info@nsslabs.com
www.nsslabs.com