NEXT GENERATION FIREWALL PRODUCT ANALYSIS

Cisco ASA 5525-X v5.3.1

Authors – Christopher Conrad, Joseph Pearce

Overview

NSS Labs performed an independent test of the Cisco ASA 5525-X v5.3.1. The product was subjected to thorough testing at the NSS facility in Austin, Texas, based on the Next Generation Firewall (NGFW) methodology v5.4 available on www.nsslabs.com. This test was conducted free of charge and NSS did not receive any compensation in return for Cisco's participation. For additional information on NGFW technology, refer to the NSS Analysis Brief entitled "*What Do CIOs Need to Know About Next Generation Firewalls?*"

While the companion Comparative Analysis Reports (CAR) on security, performance, and total cost of ownership (TCO) will provide comparative information about all tested products, this individual Product Analysis Report (PAR) 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 tune the IPS module separately within their NGFW. Therefore, NSS evaluation of NGFW products is configured with the vendor pre-defined or recommended, "out-of-the-box" settings, in order to provide readers with relevant security effectiveness and performance dimensions based upon their expected usage.

Product		Exploit Block Rate[1]		NSS-Tested Throughput
Cisco ASA 5525-X v5.3.1		99.2%		954 Mbps
Evasions	Stability & Reliability	Application Control	Identity Aware	Firewall Policy Enforcement
PASS	PASS	PASS	PASS	PASS

Figure 1 – Overall Test Results

Using the recommended policy, the ASA 5525-X blocked 99.5% of attacks against server applications, 99.0% of attacks against client applications, and 99.2% overall. The device proved effective against all evasion techniques tested. The device also passed all stability and reliability tests.

The Cisco ASA 5525-X is rated by NSS at 954 Mbps, which exceeds the vendor-claimed performance (Cisco rates this device at 650 Mbps.) NSS-Tested Throughput is calculated as an average of all the "Real-World" Protocol Mixes and the 21 KB HTTP response-based capacity tests.

[1] Exploit Block Rate is defined as the number of exploits blocked under test.

Table of Contents

Table of Figures

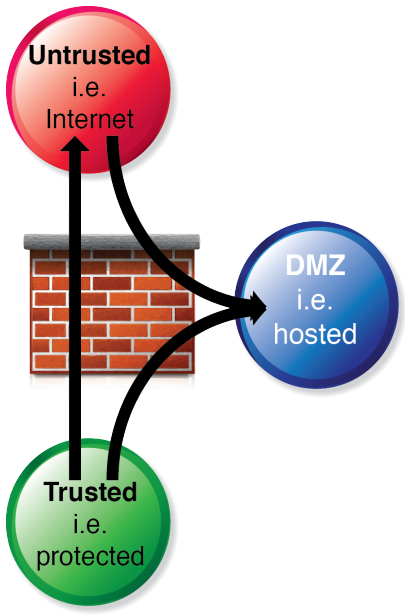
Security Effectiveness

This section verifies that the device under test (DUT) 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 a *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 an unknown and non-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; 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 a minimum 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

A NGFW must provide granular control based upon applications, not just 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 NGFW since it enables the administrator to define security policies based upon applications rather than ports alone.

Test Procedure	Result
Block Unwanted Applications	PASS
Block Specific Actions	PASS

Figure 3 – Application Control

Our testing found that the Cisco ASA 5525-X v5.3.1 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 upon the policy.

User/Group Identity (ID) Aware Policies

An NGFW should be able to identify users and groups and apply security policy based on identity. Where possible, this should be achieved via direct integration with existing enterprise authentication systems (such as Active Directory) without the need for custom server-side software. This allows the administrator to create even more granular policies.

Test Procedure	Result
Users Defined via NGFW Integration with Active Directory	PASS
Users Defined in NGFW DB (where AD integration is not available)	N/A

Figure 4 – User/Group ID Aware Policies

Integrating the ASA 5525-X with the Active Directory implementation was simple and intuitive. Testing verified that the Cisco ASA 5525-X v5.3.1 correctly enforced complex outbound and inbound policies consisting of multiple rules, objects and applications. NSS engineers verified that the device successfully identified the users and groups and took the appropriate action based upon the firewall policy.

Exploit Block Rate

In order to accurately represent the protection that may be achieved by the intrusion prevention system (IPS) module, NSS evaluates the DUT using the recommended policy. The results in this report are using the recommended configuration that ships with the product "out-of-the-box".

Live Exploit Testing: NSS' security effectiveness testing leverages the deep expertise of our engineers to generate the same types of attacks used by modern cyber criminals, utilizing multiple commercial, open source, and proprietary tools as appropriate. With over 1800 live exploits, this is the industry's most comprehensive test to date. Most notable, all of the live exploits and payloads in these tests 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
- A malicious payload is installed
- The system is rendered unresponsive
- Etc.

Product	Total Number of Exploits Run	Total Number Blocked	Block Percentage
Cisco ASA 5525-X v5.3.1	1,841	1,827	99.2%

Figure 5 – Number of Exploits Blocked in %

False Positive Testing

The Cisco ASA 5525-X v5.3.1 demonstrated adequate capability of correctly identifying traffic and did not fire IPS alerts on non-malicious content.

Coverage by Attack Vector

Because a failure to block attacks could result in significant compromise and impact to critical business systems, Network Intrusion Prevention Systems should be evaluated against a broad set of exploits. Exploits can be categorized into two groups: *attacker-initiated* and *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. In target-initated exploits, the most common type of attack experienced by the end user, the attacker has little or no control as to when the threat is executed.

	Attacker Initiated	Target Initiated
Attempted	828	1013
Caught	824	1003
Coverage	99.5%	99.0%

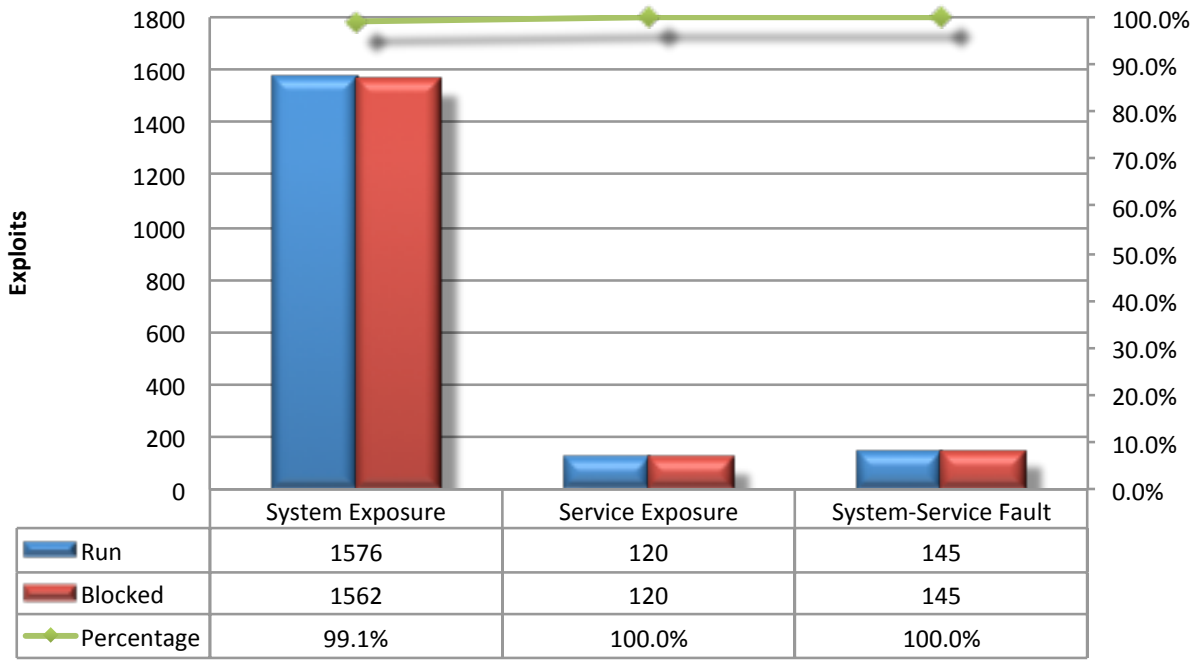
Figure 6 – Coverage by Attack Vector

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. Typical attacks in this category include service-specific attacks, such as SQL injection, that enable an attacker to execute arbitrary SQL commands within the database service. These attacks are somewhat isolated to the service and do not immediately result in full system-level access to the operating system and all services. However, by using additional localized system attacks, it may be possible for the attacker to escalate from the service level to the system level.

Finally, there are the attacks which 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. These attacks do not enable the attacker to execute arbitrary commands. Still, the resulting impact to the business could be severe, as the attacker could crash a protected system or service.



	System Exposure	Service Exposure	System-Service Fault
Run	1576	120	145
Blocked	1562	120	145
Percentage	99.1%	100.0%	100.0%

Figure 7 – Product Coverage by Impact

Coverage by Date

This graph provides insight into whether a vendor ages out protection signatures aggressively in order to preserve performance levels. It also reveals where a product lags behind in protection for the most recent vulnerabilities. NSS will report exploits by individual years for the past 10 years. Exploits older than 10 years will be consolidated into the oldest "bucket."

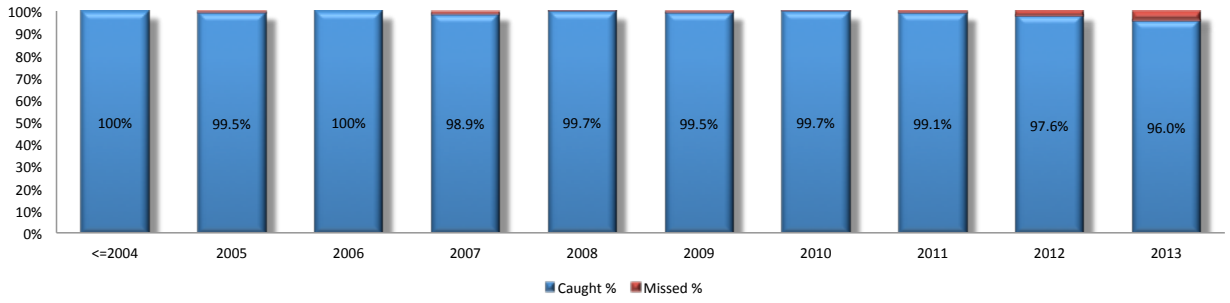


Figure 8 – Product Coverage by Date

Coverage by Target Vendor

The NSS exploit library covers a wide range of protocols and applications representing a wide range of software vendors. This graph highlights the coverage offered by the Cisco ASA 5525-X for some of the top vendor targets (out of more than 70) represented for this round of testing.

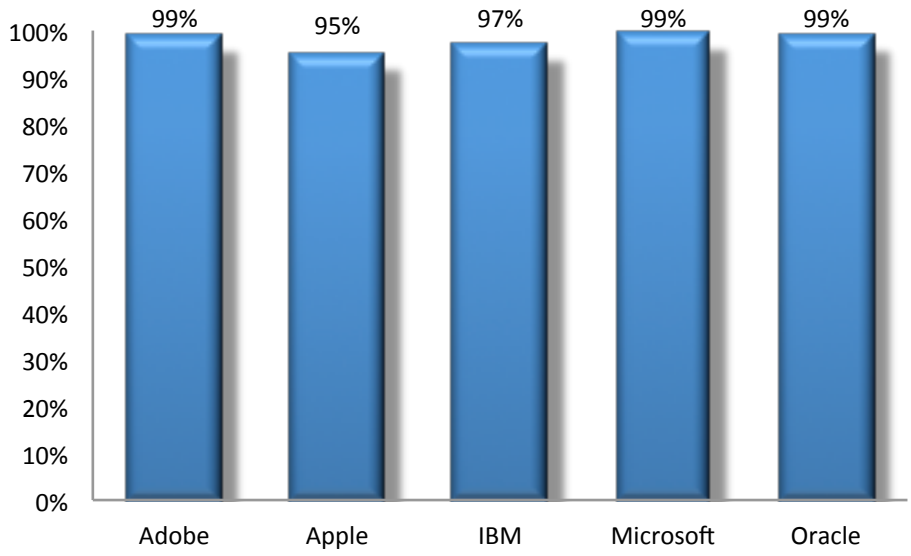


Figure 9 – Product Coverage by Target Vendor

Coverage by Result

These tests determine the protection provided against different types of exploits based on the intended action of those exploits, e.g., arbitrary execution, buffer overflow, code injection, cross-site scripting, directory traversal, privilege escalation, etc. Further details are available to NSS clients via inquiry call.

Coverage by Target Type

These tests determine the protection provided against different types of exploits based on the target environment, e.g. Web server, Web browser, database, ActiveX, Java, browser plugins, etc. Further details are available to NSS clients via inquiry call.

Resistance to Evasion Techniques

Evasion techniques are a means of disguising and modifying attacks at the point of delivery in order to avoid detection and blocking by security products. Failure of a security device to handle correctly a particular type of evasion potentially will allow 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—IP Packet Fragmentation, Stream Segmentation, RPC fragmentation, SMB & NetBIOS Evasions, URL Obfuscation, HTML Obfuscation, Payload Encoding and FTP evasion—the less effective the device.

For example, it is better to miss all techniques in one evasion category (say, 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) will have a greater impact on security effectiveness than those operating at the upper layers (HTTP or FTP obfuscation.) This is because lower-level evasions will impact potentially a wider number of exploits; therefore, missing TCP segmentation is a much more serious issue than missing FTP obfuscation.

Figure 10 provides the results of the evasion tests for Cisco ASA 5525-X.

Test Procedure	Result
IP Packet Fragmentation	PASS
Stream Segmentation	PASS
RPC Fragmentation	PASS
SMB & NetBIOS Evasions	PASS
URL Obfuscation	PASS
HTML Obfuscation	PASS
Payload Encoding	PASS
FTP Evasion	PASS
IP Packet Fragmentation + TCP Segmentation	PASS
IP Packet Fragmentation + MSRPC Fragmentation	PASS
IP Packet Fragmentation + SMB Evasions	PASS
Stream Segmentation + SMB & NETBIOS Evasions	PASS

Figure 10 – Resistance to Evasion Results

The device, when tested against all evasion techniques, successfully decoded all evasions attempted above and triggered IPS alerts on the correct exploit(s). This resulted in an overall PASS result for Cisco ASA 5525-X.

Performance

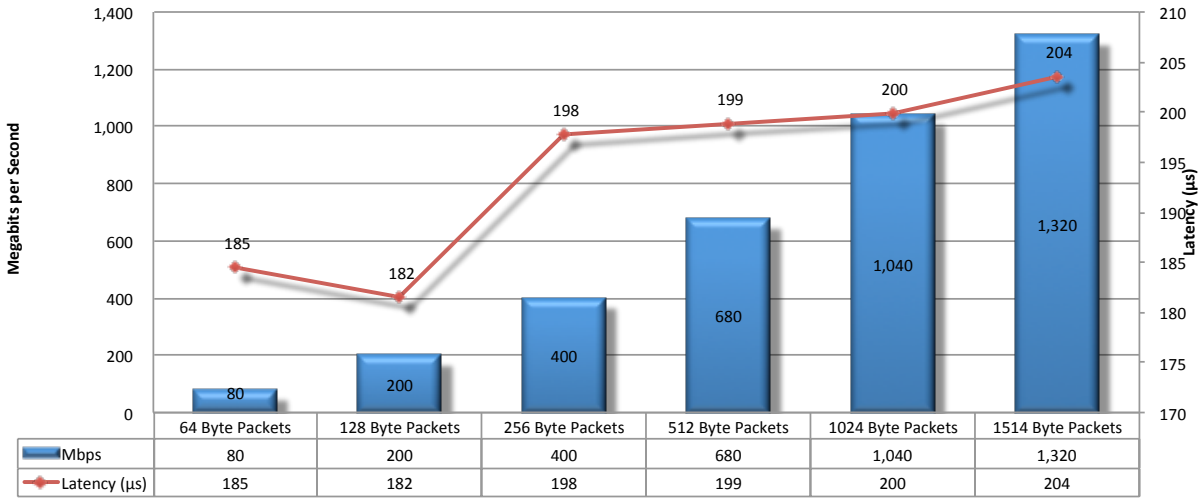
There is frequently a trade-off between security effectiveness and performance. Because of this trade-off, it is important to judge a product's security effectiveness within the context of its performance (and *vice versa*). This ensures that new security protections do not adversely impact performance and 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 bi-directionally through each port pair of the DUT.

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 in-line port pair are verified by network monitoring tools before each test begins. Multiple tests are run and averages taken where necessary.

This traffic does not attempt to simulate any form of "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 purely to determine the raw packet processing capability of each in-line port pair of the DUT, and its effectiveness at forwarding packets quickly in order to provide the highest level of network performance and lowest latency.



	64 Byte Packets	128 Byte Packets	256 Byte Packets	512 Byte Packets	1024 Byte Packets	1514 Byte Packets
Mbps	80	200	400	680	1,040	1,320
Latency (µs)	185	182	198	199	200	204

Figure 11 – Raw Packet Processing Performance (UDP Traffic)

The ASA 5525-X demonstrated a drop-off in raw packet processing performance when the frame size(s) were less than 512 bytes.

Latency – UDP

Next Generation Firewalls that introduce high levels of latency lead to unacceptable response times for users, especially where multiple security devices are placed in the data path. These results show the latency (in microseconds) as recorded during the UDP throughput tests at 90% of maximum load.

Latency - UDP	Microseconds
64 Byte Packets	185
128 Byte Packets	182
256 Byte Packets	198
512 Byte Packets	199
1024 Byte Packets	200
1514 Byte Packets	204

Figure 12 – UDP Latency in Microseconds

Connection Dynamics – Concurrency and Connection Rates

The use of sophisticated test equipment appliances allows NSS engineers to create true "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 handles 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 DUT is causing unacceptable increase in open connections on the server-side.
- **Excessive response time for HTTP transactions –** Latency within the DUT is causing excessive delays and increased response time to the client.
- **Unsuccessful HTTP transactions –** Normally, there should be zero unsuccessful transactions. Once these appear, it is an indication that excessive latency within the DUT is causing connections to time out.

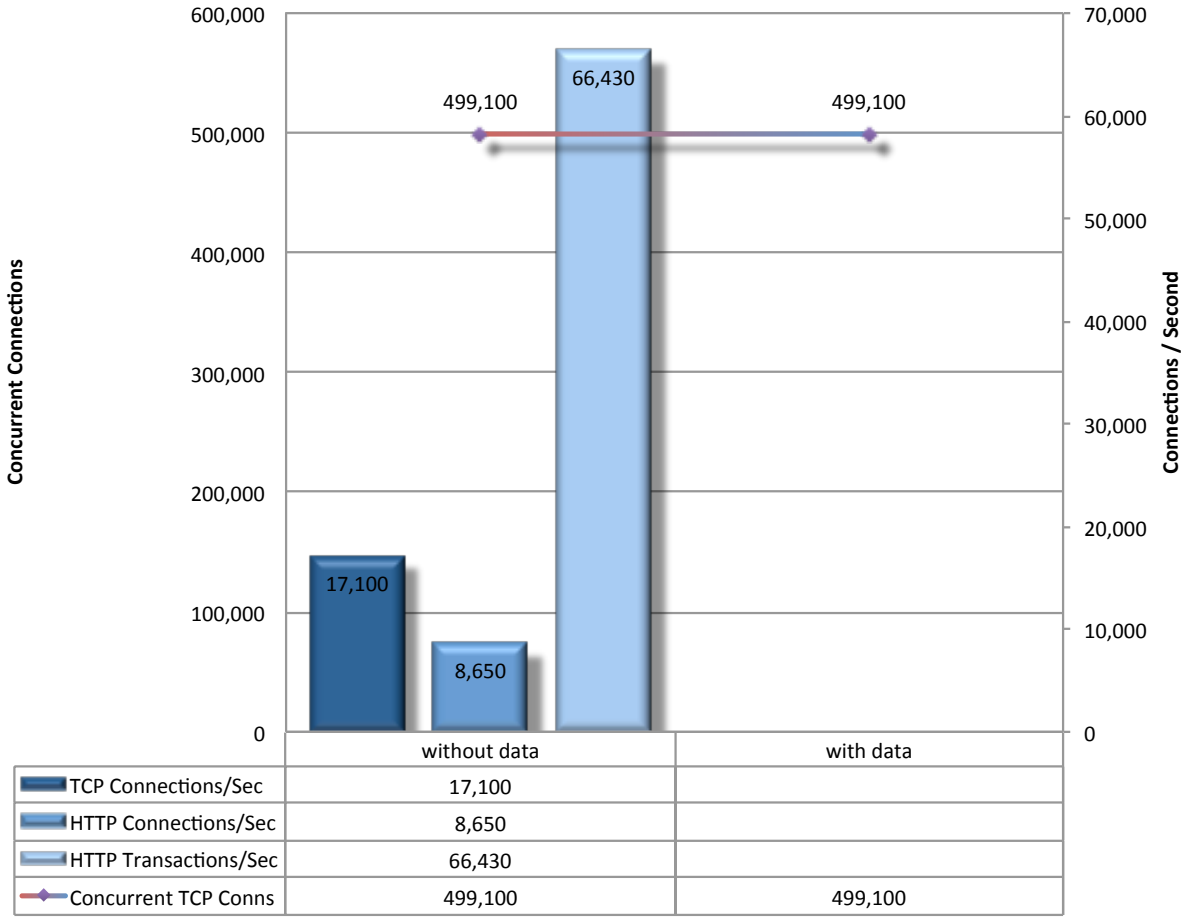


Figure 13 – Concurrency and Connection Rates

HTTP Connections per Second and Capacity

The aim of these tests is to stress the HTTP detection engine and determine how the DUT 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 DUT 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" as it is possible to achieve in a lab environment, while ensuring absolute accuracy and repeatability.

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

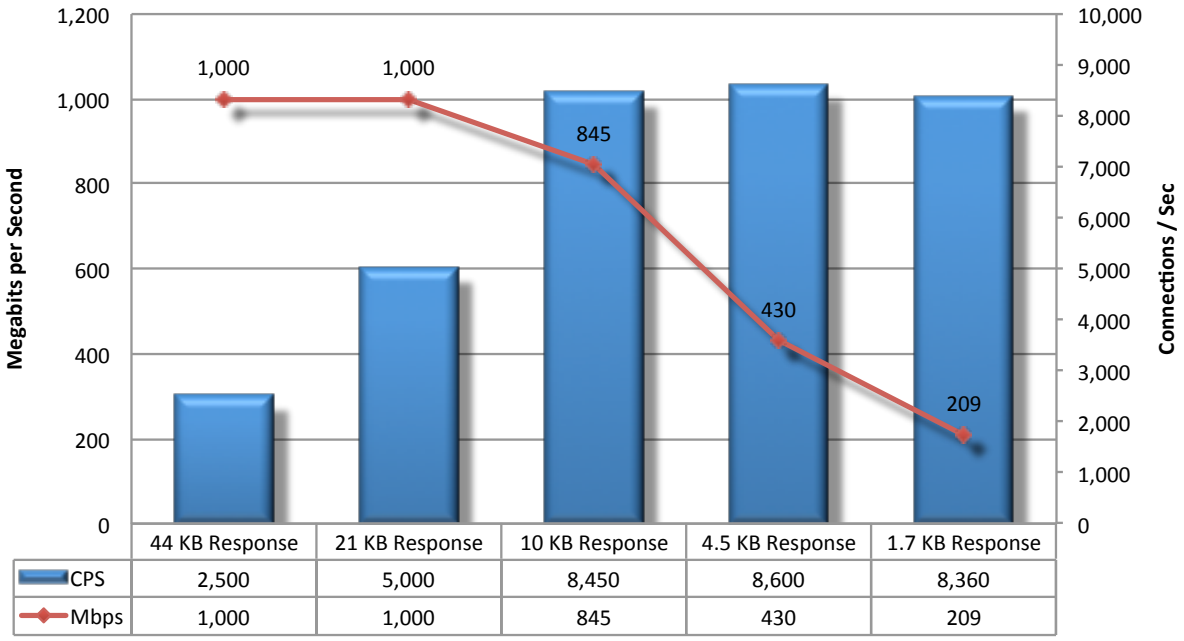


Figure 14 – HTTP Connections per Second and Capacity

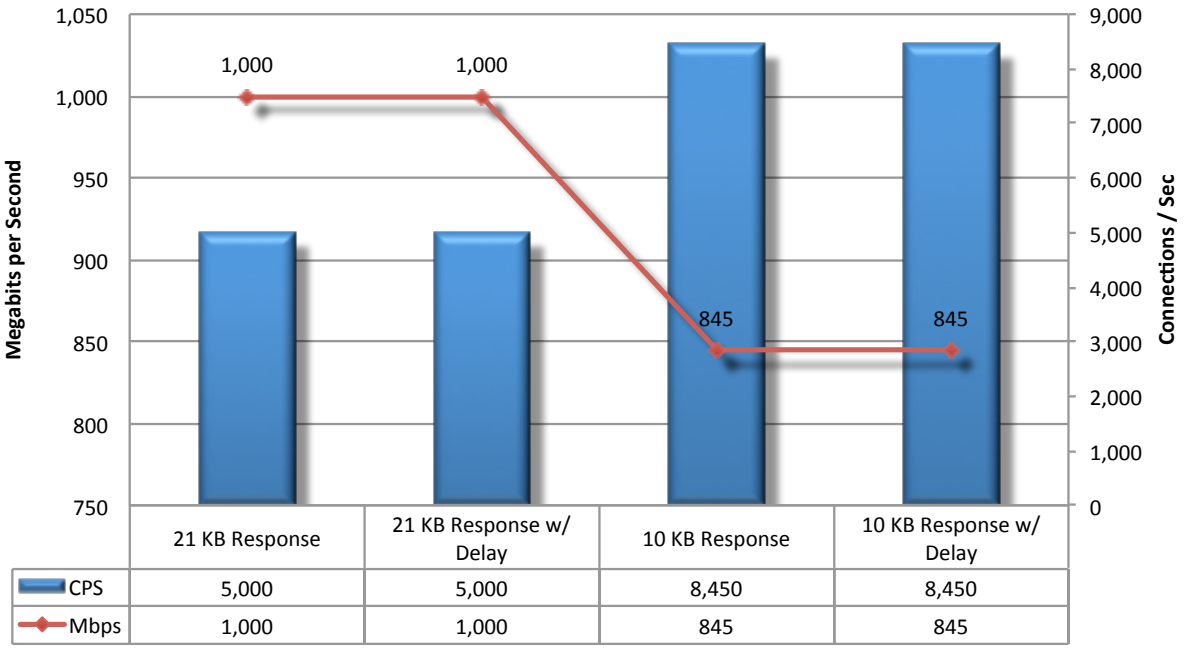
Application Average Response Time – HTTP

Application Average Response Time - HTTP (at 90% Maximum Load)	Milliseconds
2,500 Connections Per Second – 44 KB Response	1.92
5,000 Connections Per Second – 21 KB Response	1.67
10,000 Connections Per Second – 10 KB Response	1.82
20,000 Connections Per Second – 4.5 KB Response	1.73
40,000 Connections Per Second – 1.7 KB Response	1.50

Figure 15 – Average Application Response Time in Milliseconds

HTTP Connections per Second and Capacity (with Delays)

Typical user behavior introduces delays between requests and responses, e.g., "think time," as users read web pages and decide which links to click next. This group of tests is identical to the previous group except that these include a 5 second delay in the server response for each transaction. This has the effect of maintaining a high number of open connections throughout the test, thus forcing the sensor to utilize additional resources to track those connections.



	21 KB Response	21 KB Response w/ Delay	10 KB Response	10 KB Response w/ Delay
CPS	5,000	5,000	8,450	8,450
Mbps	1,000	1,000	845	845

Figure 16 –HTTP Connections per Second and Capacity (with Delays)

Real-World Traffic Mixes

This test measures the performance of the device under test 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 under test (network core or perimeter) to reflect real use cases. For details about real world traffic protocol types and percentages, see the NSS *Next Generation Firewall Test Methodology*, available at www.nsslabs.com

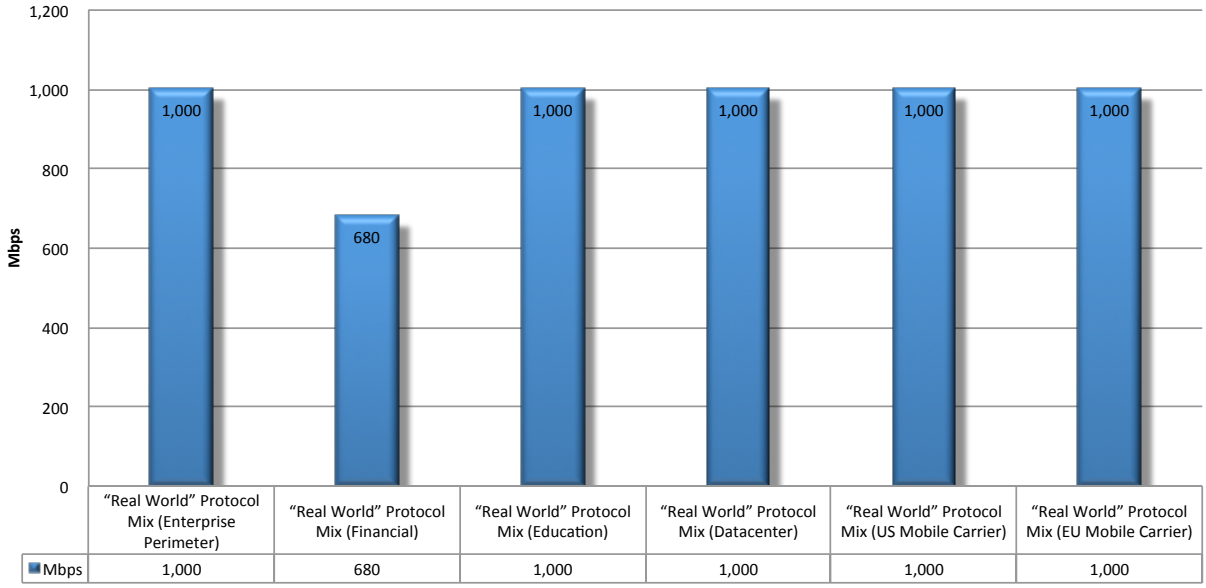


Figure 17 – Real World Traffic Mixes

The ASA 5525-X performed above the throughput claimed by the vendor with all traffic mixes.

Stability and Reliability

Long-term stability is particularly important for an in-line device, where failure can produce network outages. These tests verify the stability of the DUT along with its ability to maintain security effectiveness while under normal load and while passing malicious traffic. Products that are not able to sustain legitimate traffic (or that crash) while under hostile attack will not pass.

The ASA 5525-X 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 by either the volume of traffic or the DUT failing open for any reason, this will result in a FAIL.

Test Procedure	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 & Mutation	PASS
Power Fail	PASS
Redundancy	YES
Persistence of Data	PASS

Figure 18 – Stability and Reliability Results

These tests also determine the behavior of the state engine under load. All next generation firewall (NGFW) devices have to make the choice whether to risk denying legitimate traffic or allowing malicious traffic once they run low on resources. Dropping new connections when resources (such as state table memory) are low, or when traffic loads exceed the device capacity will theoretically block legitimate traffic, but maintain state on existing connections (preventing attack leakage).

High Availability (HA) Optional

High availability (HA) is important to many enterprise customers, and this table represents the vendors HA feature set. If no HA offering was submitted for NSS to validate, all results in this section will be marked as "N/A."

Description	Results
Failover – Legitimate Traffic	N/A
Failover – Malicious Traffic	N/A
Time to Failover	N/A
Stateful Operation	N/A
Active/Active Configuration	N/A

Figure 19 – High Availability Results

The ASA 5525-X was not configured for High Availability testing at this time.

Management and Configuration

Security devices are complicated to deploy; essential systems such as centralized management console options, log aggregation, and event correlation/management systems further complicate the purchasing decision.

Understanding key comparison points will allow customers to model the overall impact on network service level agreements (SLAs), estimate operational resource requirements to maintain and manage the systems, and better evaluate required skill / competencies of staff.

Enterprises should include management and configuration during their evaluation focusing the following at minimum:

- **General Management and Configuration –** how easy is it to install and configure devices, and deploy multiple devices throughout a large enterprise network?
- **Policy Handling –** how easy is it to create, edit, and deploy complicated security policies across an enterprise?
- **Alert Handling –** how accurate and timely is the alerting, and how easy is it to drill down to locate critical information needed to remediate a security problem?
- **Reporting –** how effective is the reporting capability, and how readily can it be customized?

Total Cost of Ownership (TCO)

Implementation of security solutions can be complex, with several factors affecting the overall cost of deployment, maintenance and upkeep. All of these 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 details the number of hours of labor required to install each device using local device management options only. This will reflect accurately the amount of time taken for NSS engineers, with the help of vendor engineers, to install and configure the DUT to the point where it operates successfully in the test harness, passes legitimate traffic and blocks/detects prohibited/malicious traffic. This closely mimics a typical enterprise deployment scenario for a single device.

Costs are based upon the time required by an experienced security engineer (assumed USD $75 per hour for the purposes of these calculations) allowing us 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)
Cisco ASA 5525-X v5.3.1	8

Figure 20 – Sensor Installation Time in Hours

Purchase Price and 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 device management (CDM) solutions may be extra. For additional TCO analysis, including CDM, refer to the *TCO CAR*.

Product	Purchase	Maintenance / year	Year 1 Cost	Year 2 Cost	Year 3 Cost	3-Year TCO
Cisco ASA 5525-X v5.3.1	$9,595	$3,171	$13,366	$3,171	$3,171	$19,709

Figure 21 – 3-Year TCO

- **Year 1 Cost** is calculated by adding installation costs ($75 USD 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**.**

This provides a TCO figure consisting of hardware, installation and maintenance costs for a single device only. Additional management and labor costs are excluded, as are TCO calculations for multiple devices, since they are modeled extensively in the *TCO CAR*.

Value: Total Cost of Ownership per Protected-Mbps

There is a clear difference between price and value. The least expensive product does not necessarily offer the greatest value if it offers significantly lower performance than only slightly more expensive competitors. The best value is a product with a low TCO and high level of secure throughput (Exploit Block Rate x NSS-Tested Throughput).

Figure 22 depicts the relative cost per unit of work performed, described as TCO per Protected-Mbps.

Product	Exploit Block Rate	NSS-Tested Throughput	3-Year TCO	TCO per Protected-Mbps
Cisco ASA 5525-X v5.3.1	99.2%	954 Mbps	$19,709	$20.81

Figure 22 – Total Cost of Ownership per Protected-Mbps

TCO per Protected-Mbps was calculated by taking the 3-Year TCO and dividing it by the product of Exploit Block Rate x NSS-Tested Throughput. Therefore 3-Year TCO/(Exploit Block Rate x NSS-Tested Throughput) = TCO per Protected-Mbps.

TCO is for single device maintenance only; costs for central device management (CDM) solutions may be extra. For additional TCO analysis, refer to the *TCO CAR*.

Detailed Product Scorecard

The following chart depicts the status of each test with quantitative results where applicable.

Description	Result
Security Effectiveness	
Firewall Policy Enforcement	
Baseline Policy	PASS
Simple Policies	PASS
Complex Policies	PASS
Static NAT	PASS
Dynamic / Hide NAT	PASS
SYN Flood Protection	PASS
IP Address Spoofing Protection	PASS
TCP Split Handshake	PASS
Application Control	
Block Unwanted Applications	PASS
Block Specific Action	PASS
User / Group ID Aware Policies	
Users Defined via NGFW Integration with Active Directory	PASS
Users Defined in NGFW DB (Alternate to 3.3.1)	N/A
Intrusion Prevention	
False Positive Testing	PASS
Coverage by Attack Vector	
Attacker Initiated	99.5%
Target Initiated	99.0%
Combined Total	99.2%
Coverage by Impact Type	
System Exposure	99.1%
Service Exposure	100%
System or Service Fault	100%
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	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

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
SMB & NetBIOS Evasions	PASS
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
A chaffed NetBIOS message is sent before the first actual NetBIOS message. The chaff message is an unspecified NetBIOS message with HTTP POST request like payload	PASS
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
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	PASS
UTF-16 character set encoding (big-endian)	PASS
UTF-16 character set encoding (little-endian)	PASS
UTF-32 character set encoding (big-endian)	PASS
UTF-32 character set encoding (little-endian)	PASS
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
Payload Encoding	PASS
x86/call4_dword_xor	PASS
x86/fnstenv_mov	PASS
x86/jmp_call_additive	PASS
x86/shikata_ga_nai	PASS
FTP 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 Packet 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 Packet 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 Packet Fragmentation + SMB & NetBIOS 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
Stream 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

Performance	
Raw Packet Processing Performance (UDP Traffic)	Mbps
64 Byte Packets	80
128 Byte Packets	200
256 Byte Packets	400
512 Byte Packets	680
1024 Byte Packets	1,040
1514 Byte Packets	1,320
Latency - UDP	Microseconds
64 Byte Packets	185
128 Byte Packets	182
256 Byte Packets	198
512 Byte Packets	199
1024 Byte Packets	200
1514 Byte Packets	204
Maximum Capacity	
Theoretical Max. Concurrent TCP Connections	499,100
Theoretical Max. Concurrent TCP Connections w/Data	499,100
Maximum TCP Connections Per Second	17,100
Maximum HTTP Connections Per Second	8,650
Maximum HTTP Transactions Per Second	66,430
HTTP Capacity with no Transaction Delays	
2,500 Connections Per Second – 44 KB Response	2,500
5,000 Connections Per Second – 21 KB Response	5,000
10,000 Connections Per Second – 10 KB Response	8,450
20,000 Connections Per Second – 4.5 KB Response	8,600
40,000 Connections Per Second – 1.7 KB Response	8,360
Application Average Response Time - HTTP (at 90% Max Load)	Milliseconds
2,500 Connections Per Second – 44 KB Response	1.92
5,000 Connections Per Second – 21 KB Response	1.67
10,000 Connections Per Second – 10 KB Response	1.82
20,000 Connections Per Second – 4.5 KB Response	1.73
40,000 Connections Per Second – 1.7 KB Response	1.50
HTTP Capacity with Transaction Delays	
21 KB Response with Delay	5,000
10 KB Response with Delay	8,450
“Real World” Traffic	Mbps
“Real World” Protocol Mix (Enterprise Perimeter)	1,000
“Real World” Protocol Mix (Financial)	680
“Real World” Protocol Mix (Education)	1,000
“Real World” Protocol Mix (Data Center)	1,000
“Real World” Protocol Mix (US Mobile Carrier)	1,000
“Real World” Protocol Mix (European Mobile Carrier)	1,000

Stability & 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 & Mutation	PASS
Power Fail	PASS
Redundancy	YES
Persistence of Data	PASS
High Availability (HA) Optional Test	
Failover - Legitimate Traffic	N/A
Failover - Malicious Traffic	N/A
Time to Failover	N/A
Stateful Operation	N/A
Active-Active Configuration	N/A
Total Cost of Ownership	
Ease of Use	
Initial Setup (Hours)	8
Time Required for Upkeep (Hours per Year)	*Contact NSS*
Expected Costs	
Initial Purchase (hardware as tested)	$9,595
Installation Labor Cost (@ USD $75/hr)	$600
Annual Cost of Maintenance & Support (hardware/software)	$1,151
Annual Cost of Updates (IPS/AV/etc.)	$2,020
Initial Purchase (enterprise management system)	*Contact NSS*
Annual Cost of Maintenance & Support (enterprise management system)	*Contact NSS*
Management Labor Cost (per Year @ USD $75/hr)	*Contact NSS*
Total Cost of Ownership	
Year 1	$13,366
Year 2	$3,171
Year 3	$3,171
3-Year Total Cost of Ownership	$19,709

Figure 23 – Detailed Scorecard

Test Methodology

Methodology Version: Next Generation Firewall Test Methodology v5.4

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 Rd
Building A, Suite 200
Austin, TX 78746
+1 (512) 961-5300
info@nsslabs.com
www.nsslabs.com