NEXT GENERATION INTRUSION PREVENTION SYSTEM (NGIPS) TEST REPORT

Palo Alto Networks PA-5020 PAN-OS v6.1.1

Author – Ty Smith

Overview

NSS Labs performed an independent test of the Palo Alto Networks PA-5020 PAN-OS v6.1.1. The product was subjected to thorough testing at the NSS facility in Austin, Texas, based on the *Next Generation Intrusion Prevention System (NGIPS) Methodology v1.0* available at www.nsslabs.com. This test was conducted free of charge and NSS did not receive any compensation in return for Palo Alto Networks' participation.

While the companion comparative reports on security, performance, and total cost of ownership (TCO) will provide comparative information about all tested products, this individual test report provides detailed information not available elsewhere.

NSS research indicates that the majority of enterprises tune their NGIPS. Therefore, NSS' evaluates NGIPS products as optimally tuned by the vendor prior to testing. Every effort is made to deploy policies that ensure the optimal combination of security effectiveness and performance, as would be the aim of a typical customer deploying the device in a live network environment.

Product	Exploit Block Rate	NSS-Tested Throughput
Palo Alto Networks PA-5020 PAN-OS v6.1.1	98.8%[1]	2,973 Mbps

Evasions	Stability and Reliability
PASS	PASS

Figure 1 – Overall Test Results (Tuned Policies)

Using a tuned policy, the PA-5020 blocked 98.8% of exploits. The device proved effective against all evasion techniques tested. The device also passed all stability and reliability tests.

The Palo Alto Networks PA-5020 is rated by NSS at 2,973 Mbps, which is higher than the vendor-claimed performance; Palo Alto Networks rates this device at 2Gbps. 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] The exploit block rate is defined as the percentage of exploits and live (real-time) drive-by 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.

Exploit Library

In order to accurately represent the protection that may be achieved, NSS evaluates the DUT using a tuned policy.

Exploit Testing: NSS' *security effectiveness* testing leverages the deep expertise of NSS engineers to generate the same types of attacks used by modern cybercriminals, utilizing multiple commercial, open-source, and proprietary tools as appropriate. With over 1800 exploits, this is the industry's most comprehensive test to date. Most notable, all of the 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
Palo Alto Networks PA-5020 PAN-OS v6.1.1	1898	1852	97.6%

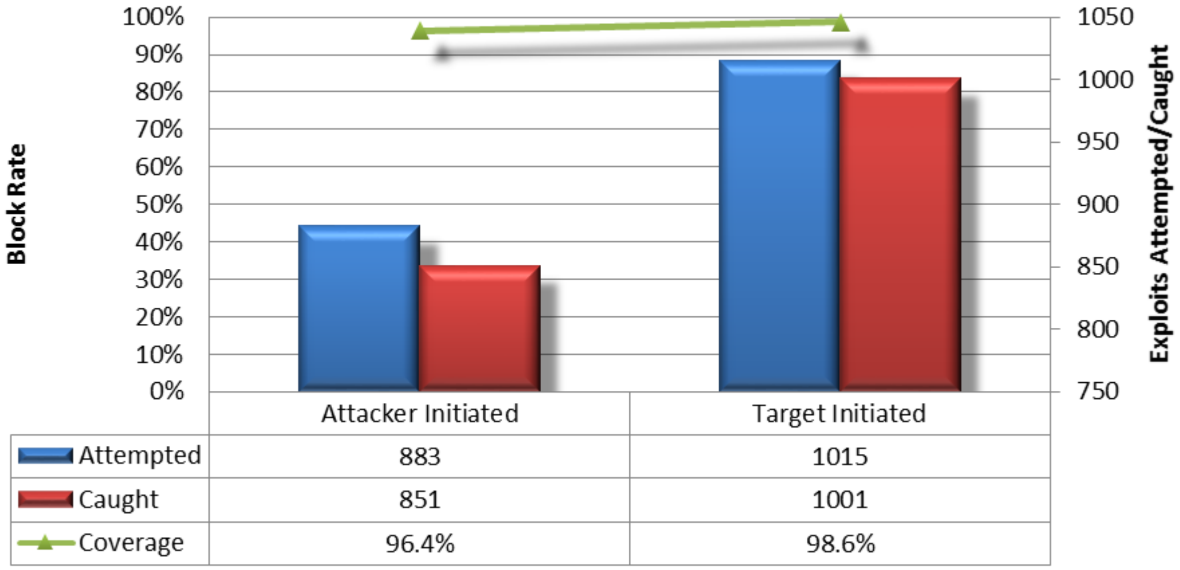
Figure 2 – Number of Exploits Blocked in %

False Positive Testing

The Palo Alto Networks PA-5020 PAN-OS v6.1.1 correctly identified traffic and did not fire IPS alerts for 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. With 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	883	1015
🟥 Caught	851	1001
▲ Coverage	96.4%	98.6%

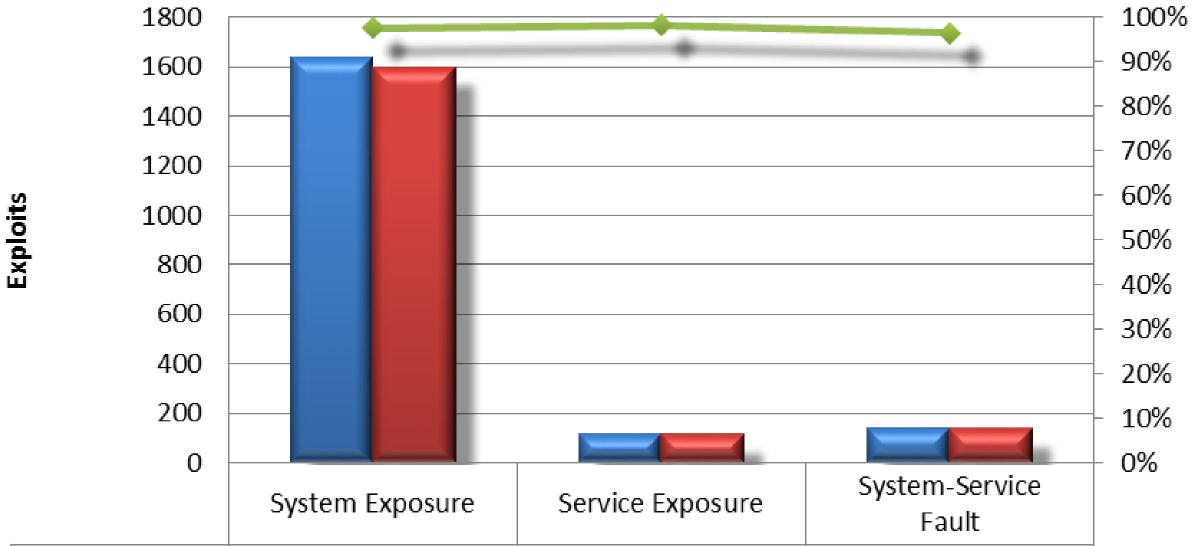
Figure 3 – Coverage by Attack Vector

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	1639	117	142
🟥 Blocked	1600	115	137
◆ Percentage	97.6%	98.3%	96.5%

Figure 4 – 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.”

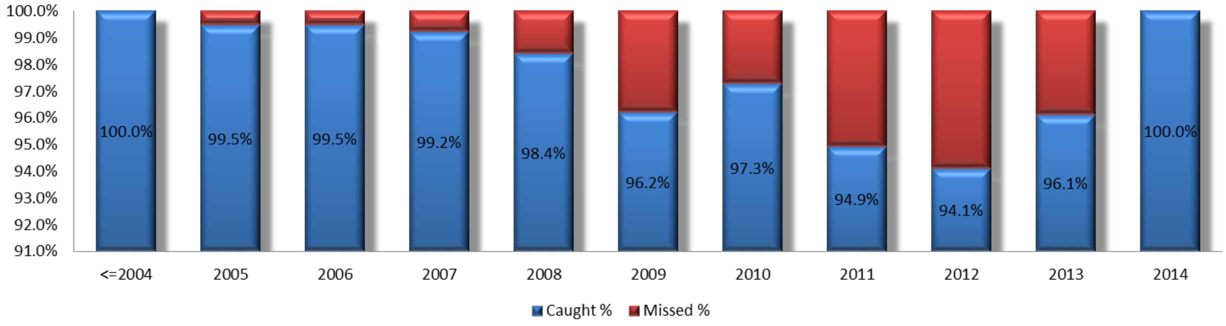


Figure 5 – Product Coverage by Date

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 Palo Alto Networks PA-5020 for some of the top vendor targets (out of more than 70) represented for this round of testing.

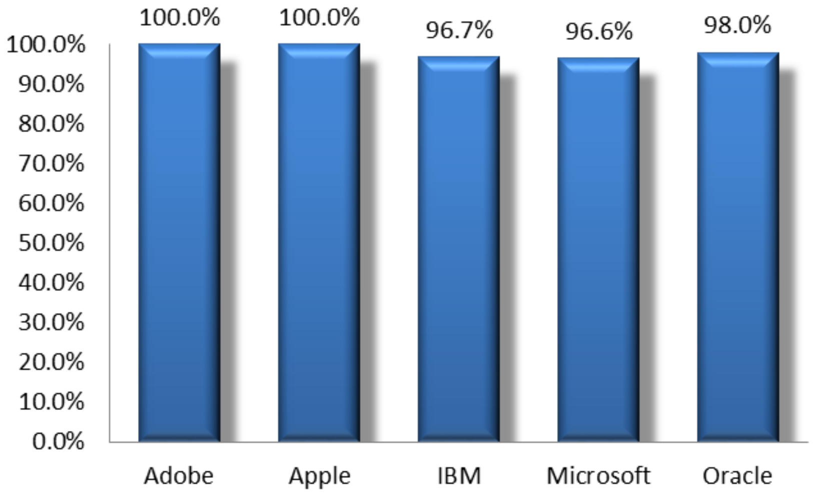


Figure 6 – Product Coverage by Target Vendor

Coverage by Result

These tests determine the protection provided against different types of exploits based on the intended action of those exploits, for example, arbitrary execution, buffer overflow, code injection, cross-site scripting, directory traversal, or privilege escalation. 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, for example, web server, web browser, database, ActiveX, Java, browser plugins, etc. Further details are available to NSS clients via inquiry call.

Live (Real-Time) Drive-by Exploits

While the NSS exploit library covers diverse protocols and applications representing a wide range of software vendors (broad coverage), the live (real-time) drive-by exploits focus on current threats (live coverage).[2] Protection from web-based exploits targeting client applications, also known as "drive-by" downloads, can be effectively measured in the NSS unique live test harness through a series of procedures that measures the stages of protection.

Unlike traditional malware that is downloaded and installed, "drive-by" attacks first exploit a vulnerable application and then silently download and install malware. This means that there are three opportunities to break the chain of events leading to a successful compromise:

1. URL access (reputation)
2. Exploit
3. Malware

To test vendors' ability to block current threats, NSS collects real threats and attack methods that cyber criminals and other threat actors use against the NSS global threat intelligence network.

Success or failure is determined based on whether the device blocks the attack. Attacks that are not successfully blocked will be measured as a failure.

Figure 7 depicts the block percentage for live drive-by exploits.

Product	Total Number of Live Exploits	Total Number Blocked	Block Percentage
Palo Alto Networks PA-5020 PAN-OS v6.1.1	613	613	100%

Figure 7— Number of Live Exploits Blocked in %

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

Application Control (Optional Test)

An NGIPS should 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 ports traditionally used by common and pervasive protocols such as HTTP/S. As such, granular application control is a requirement of an NGIPS since it enables the administrator to define security policies based upon applications rather than ports alone. Figure 8 depicts whether Palo Alto Networks PA-5020 passed or failed the application control test. Demonstration of application control functionality is optional for version 1.0 of the NGIPS methodology. Vendors that opt out of this test will be marked as "N/A."

Test Procedure	Result
Block Unwanted Applications	PASS

Figure 8 – Application Control

NSS engineers verified that Palo Alto Networks PA-5020 PAN-OS v6.1.1 successfully identified and blocked a specific application as configured.

User/Group Identity (ID) Aware Policies (Optional Test)

An NGIPS 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. Figure 9 depicts whether Palo Alto Networks PA-5020 passed or failed the user/group ID test. Demonstration of user/group aware policy functionality is optional for version 1.0 of the NGIPS methodology. Vendors that opt out of this test will be marked as "N/A."

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

Figure 9 – User/Group ID Aware Policies

NSS engineers verified that the Palo Alto Networks PA-5020 PAN-OS v6.1.1 successfully enforced user-aware policies as configured.

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 correctly handle 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 NGIPS 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 and 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 potentially impact 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 Palo Alto Networks PA-5020.

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

Figure 10 – Resistance to Evasion Results

Performance

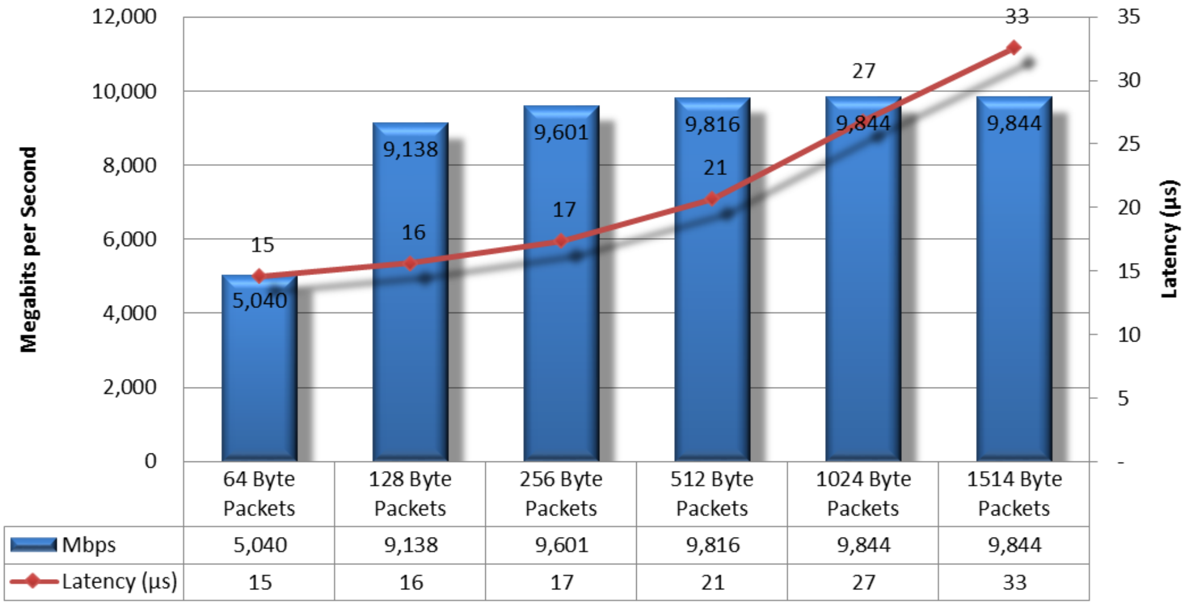
There is frequently a trade-off between security effectiveness and performance. Because of this trade-off, it is important to judge a product's security effectiveness within the context of its performance (and vice versa). This ensures that new security protections do not adversely impact performance and 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	5,040	9,138	9,601	9,816	9,844	9,844
Latency (µs)	15	16	17	21	27	33

Figure 11 – Raw Packet Processing Performance (UDP Traffic)

Latency – UDP

Next generation intrusion prevention systems that introduce high levels of latency lead to unacceptable response times for users, particulary 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	15
128 Byte Packets	16
256 Byte Packets	17
512 Byte Packets	21
1024 Byte Packets	27
1514 Byte Packets	33

Figure 12 – UDP Latency in Microseconds

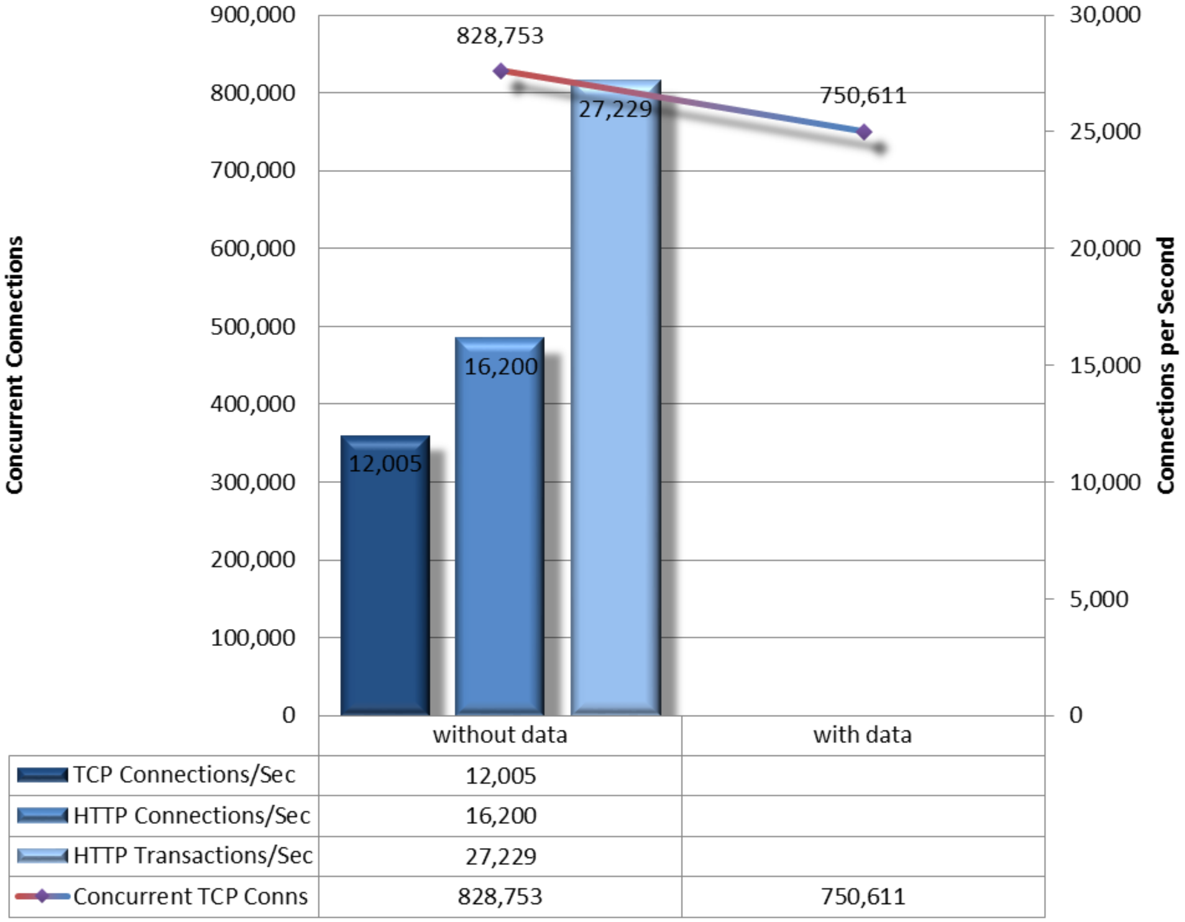
Connection Dynamics – Concurrency and Connection Rates

The use of sophisticated test equipment appliances allows NSS engineers to simulate 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.



	without data	with data
TCP Connections/Sec	12,005	
HTTP Connections/Sec	16,200	
HTTP Transactions/Sec	27,229	
Concurrent TCP Conns	828,753	750,611

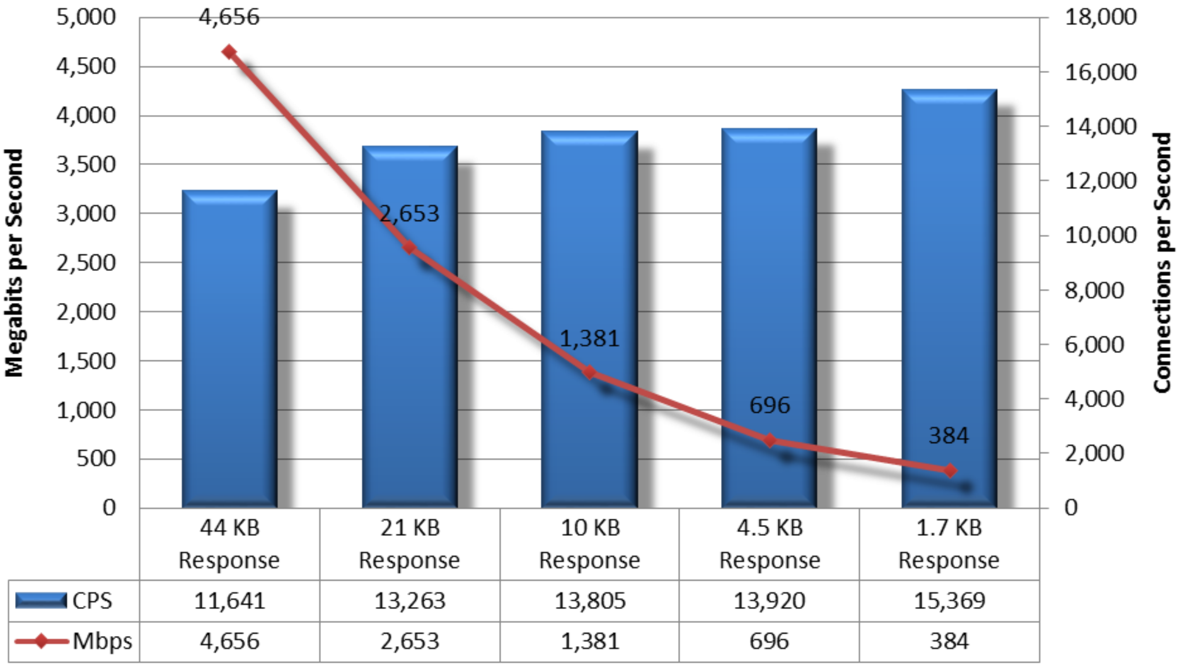
Figure 13 – Concurrency and Connection Rates

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.

HTTP Capacity with No Transaction Delays

Each transaction consists of a single HTTP GET request and there are no transaction delays (that is, 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.

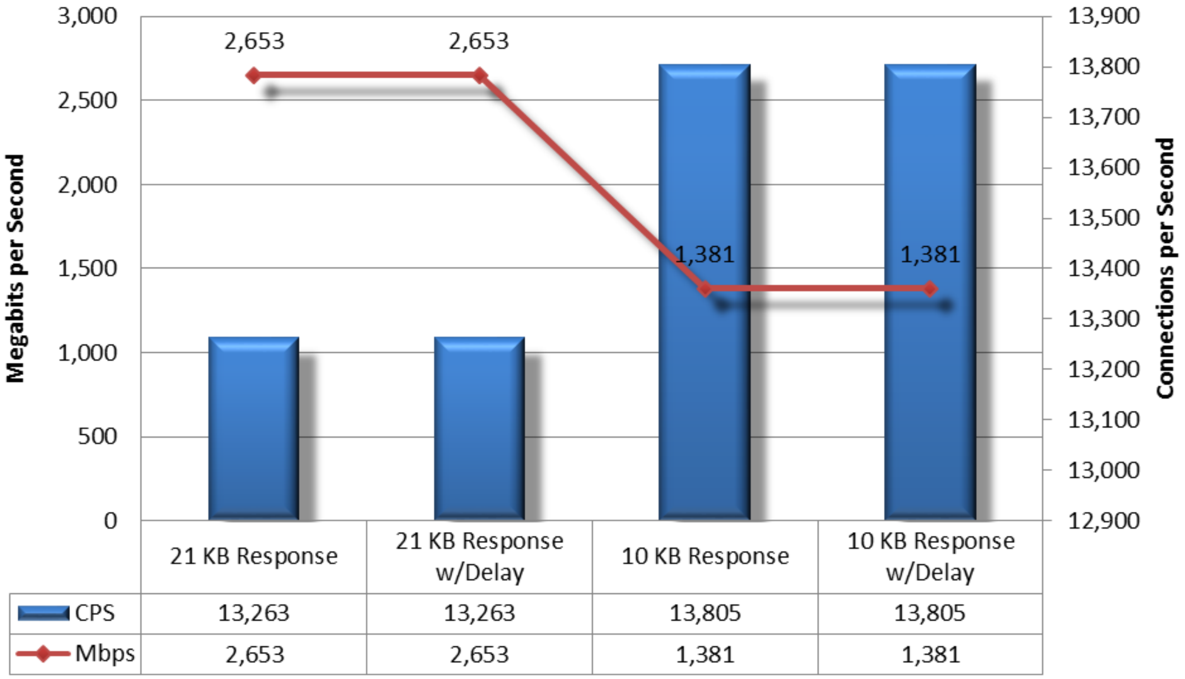


	44 KB Response	21 KB Response	10 KB Response	4.5 KB Response	1.7 KB Response
▬ CPS	11,641	13,263	13,805	13,920	15,369
◆ Mbps	4,656	2,653	1,381	696	384

Figure 14 – HTTP Connections per Second and Capacity

HTTP Capacity with Transaction Delays

Typical user behavior introduces delays between requests and reponses, for example, “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	13,263	13,263	13,805	13,805
Mbps	2,653	2,653	1,381	1,381

Figure 15 – HTTP Capacity with Transaction Delays

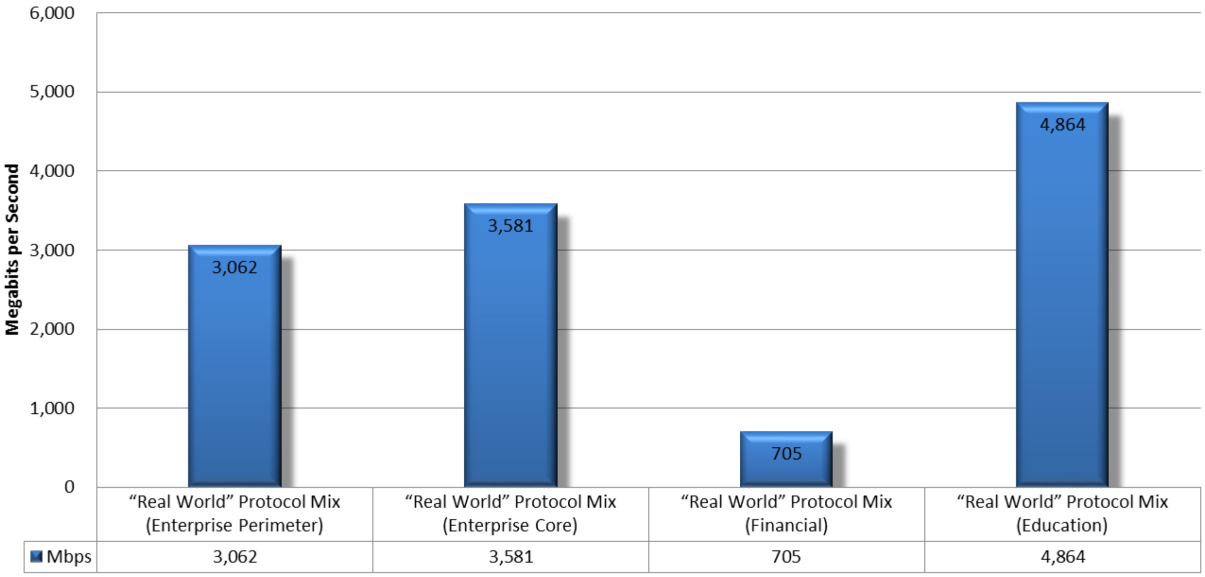
Application Average Response Time – HTTP

Application Average Response Time – HTTP (at 90% Maximum Load)	Milliseconds
2,500 Connections Per Second – 44 KB Response	14.86
5,000 Connections Per Second – 21 KB Response	8.68
10,000 Connections Per Second – 10 KB Response	5.33
20,000 Connections Per Second – 4.5 KB Response	5.92
40,000 Connections Per Second – 1.7 KB Response	11.45

Figure 16 – Average Application Response Time in Milliseconds

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 *Next Generation Intrusion Prevention System (NGIPS) Methodology v1.0* available at www.nsslabs.com.



	“Real World” Protocol Mix (Enterprise Perimeter)	“Real World” Protocol Mix (Enterprise Core)	“Real World” Protocol Mix (Financial)	“Real World” Protocol Mix (Education)
■ Mbps	3,062	3,581	705	4,864

Figure 17 – Real World Traffic Mixes

The PA-5020 performed above vendor-claimed throughput claimed for all of the “real-world” mixes with the exception of the financial mix.

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 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 by either the volume of traffic or the DUT failing open for any reason, the device will fail the test.

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 –Detection Port	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 NGIPS devices must choose 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).

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 on the following at a 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 items are included for a single device only (the cost of acquisition and installation).

Installation (Hours)

This table depicts the amount of time that NSS engineers, with the help of vendor engineers, needed 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. For purposes of this test report, a rate of US$75 per hour was used. Clients can substitute their own installation time estimates and labor costs to obtain accurate TCO figures.

Product	Installation (Hours)
Palo Alto Networks PA-5020 PAN-OS v6.1.1	8

Figure 19 – 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 management solutions (CMS) may be extra. For additional TCO analysis, including CMS, refer to the *TCO Comparative Report*.

Product	Purchase	Maintenance / Year	Year 1 Cost	Year 2 Cost	Year 3 Cost	3-Year TCO
Palo Alto Networks PA-5020 PAN-OS v6.1.1	$41,500	$10,240	$52,340	$10,240	$10,240	$72,820

Figure 20 – 3-Year TCO

- **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**.**

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 Comparative Report*.

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 21 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
Palo Alto Networks PA-5020 PAN-OS v6.1.1	98.8%	2,973	$72,820	$25

Figure 21 – 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 CMS may be extra. For additional TCO analysis, including CMS, refer to the *TCO Comparative Report*.

Detailed Product Scorecard

The following chart depicts the status of each test with quantitative results where applicable.	
Security Effectiveness	
Exploit Library and Live (Real-Time) Drive-by Exploits	98.8%
Intrusion Prevention Policies	
False Positive Testing	PASS
Coverage by Attack Vector	
Attacker Initiated	96.4%
Target Initiated	98.6%
Combined Total (Exploit Library)	**97.6%**
Coverage by Impact Type	
System Exposure	97.6%
Service Exposure	98.3%
System or Service Fault	96.5%
Coverage by Date	Contact NSS
Coverage by Target Vendor	Contact NSS
Coverage by Result	Contact NSS
Coverage by Target Type	Contact NSS
Live (Real-Time) Drive-by Exploits	
Live Exploits Blocked	**100.0%**
Application Control (Optional)	
Block Unwanted Applications	PASS
User / Group ID Aware Policies (Optional)	
Users Defined via NGIPS Integration with Active Directory	PASS
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
TCP Stream Segmentation	PASS
Ordered 1-byte segments, interleaved duplicate segments with invalid TCP checksums	PASS
Ordered 1-byte segments, interleaved duplicate segments with null TCP control flags	PASS
Ordered 1-byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream	PASS
Ordered 1-byte segments, duplicate last packet	PASS
Ordered 2-byte segments, segment overlap (favor new)	PASS
Ordered 1-byte segments, interleaved duplicate segments with out-of-window sequence numbers	PASS
Out of order 1-byte segments	PASS
Out of order 1-byte segments, interleaved duplicate segments with faked retransmits	PASS
Ordered 1-byte segments, segment overlap (favor new)	PASS
Out of order 1-byte segments, PAWS elimination (interleaved duplicate segments with older TCP timestamp options)	PASS
Ordered 16-byte segments, segment overlap (favor new (Unix))	PASS
Ordered 32-byte segments	PASS
Ordered 64-byte segments	PASS
Ordered 128-byte segments	PASS
Ordered 256-byte segments	PASS
Ordered 512-byte segments	PASS
Ordered 1024-byte segments	PASS
Ordered 2048-byte segments (sending MSRPC request with exploit)	PASS
Reverse Ordered 256-byte segments, segment overlap (favor new) with random data	PASS
Reverse Ordered 512-byte segments, segment overlap (favor new) with random data	PASS
Reverse Ordered 1024-byte segments, segment overlap (favor new) with random data	PASS
Reverse Ordered 2048-byte segments, segment overlap (favor new) with random data	PASS
Out of order 1024-byte segments, segment overlap (favor new) with random data, Initial TCP sequence number is set to 0xffffffff - 4294967295	PASS
Out of order 2048-byte segments, segment overlap (favor new) with random data, Initial TCP sequence number is set to 0xffffffff - 4294967295	PASS
RPC Fragmentation	PASS
One-byte fragmentation (ONC)	PASS
Two-byte fragmentation (ONC)	PASS
All fragments, including Last Fragment (LF) will be sent in one TCP segment (ONC)	PASS
All frags except Last Fragment (LF) will be sent in one TCP segment. LF will be sent in separate TCP seg (ONC)	PASS
One RPC fragment will be sent per TCP segment (ONC)	PASS
One LF split over more than one TCP segment. In this case no RPC fragmentation is performed (ONC)	PASS
Canvas Reference Implementation Level 1 (MS)	PASS
Canvas Reference Implementation Level 2 (MS)	PASS
Canvas Reference Implementation Level 3 (MS)	PASS
Canvas Reference Implementation Level 4 (MS)	PASS
Canvas Reference Implementation Level 5 (MS)	PASS
Canvas Reference Implementation Level 6 (MS)	PASS
Canvas Reference Implementation Level 7 (MS)	PASS
Canvas Reference Implementation Level 8 (MS)	PASS
Canvas Reference Implementation Level 9 (MS)	PASS

Canvas Reference Implementation Level 10 (MS)	PASS
MSRPC messages are sent in the big endian byte order, 16 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 32 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 64 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 128 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 256 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 512 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
MSRPC messages are sent in the big endian byte order, 1024 MSRPC fragments are sent in the same lower layer message, MSRPC requests are fragmented to contain at most 2048 bytes of payload	PASS
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
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
Payload Encoding	PASS
x86/call4_dword_xor	PASS
x86/fnstenv_mov	PASS
x86/jmp_call_additive	PASS
x86/shikata_ga_nai	PASS
Layered Evasions	PASS
IP Fragmentation + TCP Segmentation	PASS
Ordered 8 byte fragments + Ordered TCP segments except that the last segment comes first	PASS
Ordered 24 byte fragments + Ordered TCP segments except that the last segment comes first	PASS
Ordered 32 byte fragments + Ordered TCP segments except that the last segment comes first	PASS
Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Reverse order TCP segments, segment overlap (favor new), Overlapping data is set to zero bytes	PASS
Ordered 16 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to zero bytes	PASS
Ordered 24 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to zero bytes	PASS
Ordered 32 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to zero bytes	PASS
Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random alphanumeric	PASS
Ordered 16 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random alphanumeric	PASS
Ordered 32 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random alphanumeric	PASS

Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random bytes	PASS
Ordered 16 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random bytes	PASS
Ordered 24 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random bytes	PASS
Ordered 32 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has random payload + Out of order TCP segments, segment overlap (favor new), Overlapping data is set to random bytes	PASS
IP Fragmentation + MSRPC Fragmentation	PASS
Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 8 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 2048 bytes of payload.	PASS
Ordered 16 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 16 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 2048 bytes of payload.	PASS
Ordered 32 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 32 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 64 bytes of payload.	PASS
Ordered 64 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a shuffled payload + MSRPC messages are sent in the big endian byte order with 64 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 64 bytes of payload.	PASS
Ordered 128 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + MSRPC messages are sent in the big endian byte order with 1024 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 128 bytes of payload.	PASS
Ordered 256 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + MSRPC messages are sent in the big endian byte order with 1024 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 256 bytes of payload.	PASS
Ordered 512 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + MSRPC messages are sent in the big endian byte order with 1024 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 512 bytes of payload.	PASS
Ordered 1024 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + MSRPC messages are sent in the big endian byte order with 1024 MSRPC fragments sent in the same lower layer message. MSRPC requests are fragmented to contain at most 1024 bytes of payload.	PASS
IP Fragmentation + SMB Evasions	PASS
Ordered 1024 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + SMB chaff message before real messages. The chaff is a WriteAndX message with a broken write mode flag, and has random MSRPC request-like payload	PASS
Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + A chaffed NetBIOS message is sent before the first actual NetBIOS message. The chaff message is an unspecified NetBIOS message with MSRPC request like payload	PASS

Ordered 8 byte fragments, duplicate packet with an incrementing DWORD in the options field. The duplicate packet has a random payload + A chaffed NetBIOS message is sent before the first actual NetBIOS message. The chaff message is an unspecified NetBIOS message with HTTP GET request like payload	PASS
TCP Segmentation + SMB / NETBIOS Evasions	PASS
Reverse Ordered 2048 byte TCP segments, segment overlap (favor new) with random data + A chaffed NetBIOS message is sent before the first actual NetBIOS message. The chaff message is an unspecified NetBIOS message with MSRPC request like payload	PASS
TCP Split Handshake	PASS
Performance	
Raw Packet Processing Performance (UDP Traffic)	Mbps
64 Byte Packets	5,040
128 Byte Packets	9,138
256 Byte Packets	9,601
512 Byte Packets	9,816
1024 Byte Packets	9,844
1514 Byte Packets	9,844
Latency - UDP	Microseconds
64 Byte Packets	15
128 Byte Packets	16
256 Byte Packets	17
512 Byte Packets	21
1024 Byte Packets	27
1514 Byte Packets	33
Maximum Capacity	
Theoretical Max. Concurrent TCP Connections	828,753
Theoretical Max. Concurrent TCP Connections w/Data	750,611
Maximum TCP Connections Per Second	12,005
Maximum HTTP Connections Per Second	16,200
Maximum HTTP Transactions Per Second	27,229
HTTP Capacity With No Transaction Delays	
2,500 Connections Per Second – 44Kbyte Response	11,641
5,000 Connections Per Second – 21Kbyte Response	13,263
10,000 Connections Per Second – 10Kbyte Response	13,805
20,000 Connections Per Second – 4.5Kbyte Response	13,920
40,000 Connections Per Second – 1.7Kbyte Response	15,369
Application Average Response Time - HTTP (at 90% Max Load)	Milliseconds
2.500 Connections Per Second – 44Kbyte Response	14.86
5,000 Connections Per Second – 21Kbyte Response	8.68
10,000 Connections Per Second – 10Kbyte Response	5.33
20,000 Connections Per Second – 4.5Kbyte Response	5.92
40,000 Connections Per Second – 1.7Kbyte Response	11.45
HTTP CPS & Capacity With Transaction Delays	
21 Kbyte Response With Delay	13,263
10 Kbyte Response With Delay	13,805
“Real World” Traffic	Mbps
“Real World” Protocol Mix (Enterprise Perimeter)	3,062
“Real World” Protocol Mix (Enterprise Core)	3,581

“Real World” Protocol Mix (Financial)	705
“Real World” Protocol Mix (Education)	4,864
Stability & Reliability	
Blocking Under Extended Attack	PASS
Passing Legitimate Traffic Under Extended Attack	PASS
Behavior Of The State Engine Under Load	PASS
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
Total Cost of Ownership	
Ease of Use	
Initial Setup (Hours)	8
Time Required for Upkeep (Hours per Year)	*Contact NSS*
Time Required to Tune (Hours per Year)	*Contact NSS*
Expected Costs	
Initial Purchase (hardware as tested)	$41,500
Installation Labor Cost (@$75/hr)	$600
Annual Cost of Maintenance & Support (hardware/software)	$3,840
Annual Cost of Updates (IPS/AV/etc.)	$6,400
Initial Purchase (centralized management system)	*Contact NSS*
Annual Cost of Maintenance & Support (centralized management system)	*Contact NSS*
Management Labor Cost (per Year @$75/hr)	*Contact NSS*
Tuning Labor Cost (per Year @$75/hr)	*Contact NSS*
Total Cost of Ownership	
Year 1	$52,340
Year 2	$10,240
Year 3	$10,240
3 Year Total Cost of Ownership	$72,820

Figure 22 – Detailed Scorecard

Test Methodology

Next Generation Intrusion Prevention System: v1.0

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
info@nsslabs.com
www.nsslabs.com

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

Please read the disclaimer in this box because it contains important information that binds you. If you do not agree to these conditions, you should not read the rest of this report but should instead return the report immediately to us. “You” or “your” means the person who accesses this report and any entity on whose behalf he/she has obtained this report.

1. The information in this report is subject to change by us without notice, and we disclaim any obligation to update it.

2. The information in this report is believed by us to be accurate and reliable at the time of publication, but is not guaranteed. All use of and reliance on this report are at your sole risk. We are not liable or responsible for any damages, losses, or expenses of any nature whatsoever arising from any error or omission in this report.

3. NO WARRANTIES, EXPRESS OR IMPLIED ARE GIVEN BY US. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, ARE HEREBY DISCLAIMED AND EXCLUDED BY US. IN NO EVENT SHALL WE BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, PUNITIVE, EXEMPLARY, OR INDIRECT DAMAGES, OR FOR ANY LOSS OF PROFIT, REVENUE, DATA, COMPUTER PROGRAMS, OR OTHER ASSETS, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

4. This report does not constitute an endorsement, recommendation, or guarantee of any of the products (hardware or software) tested or the hardware and/or software used in testing the products. The testing does not guarantee that there are no errors or defects in the products or that the products will meet your expectations, requirements, needs, or specifications, or that they will operate without interruption.

5. This report does not imply any endorsement, sponsorship, affiliation, or verification by or with any organizations mentioned in this report.

6. All trademarks, service marks, and trade names used in this report are the trademarks, service marks, and trade names of their respective owners.