NEXT GENERATION INTRUSION PREVENTION SYSTEM (NGIPS) TEST REPORT

Fortinet FortiGate 3000D v5.6.4GA build 7892

SEPTEMBER 20, 2018

Authors – Thomas Williams, Matthew Wheeler, Tim Otto

Overview

NSS Labs performed an independent test of the Fortinet FortiGate 3000D v5.6.4GA build 7892. The product was subjected to thorough testing at the NSS facility in Austin, Texas, based on the Next Generation Intrusion Prevention System (NGIPS) Test Methodology v4 and the NSS Labs Evasions Test Methodology v1.1, both available at www.nsslabs.com. Testing was conducted free of charge and NSS did not receive any compensation in return for Fortinet's participation.

While the companion Comparative Reports on security, performance, and total cost of ownership (TCO) will provide information about all tested products, this Test Report provides detailed information not available elsewhere.

Vendor-Provided Settings

NGIPS products are deployed at the corporate network perimeter as well as within the network to protect employee desktops, laptops, and PCs. NSS research has determined that the majority of enterprises do not tune their NGIPS products, but rather rely on a vendor's default/recommended policies and settings. This product was tested using vendor-provided settings, i.e., the signatures/filters/rules that trigger false positives were turned off in order to replicate an enterprise environment. This Test Report provides results of the product tested as configured and submitted by the vendor.

Product	NSS-Tested Throughput		3-Year TCO (US$)
Fortinet FortiGate 3000D v5.6.4GA build 7892	15,848 Mbps		$88,725
	Exploit Block Rate[1]	Evasions Blocked[2]	Stability & Reliability
	99.6%	147/147	PASS

Figure 1 – Overall Test Results

Using the vendor-provided settings, the FortiGate 3000D blocked 99.6% of attacks. The device proved effective against 147 out of 147 evasions tested. The device passed all stability and reliability tests.

The FortiGate 3000D is rated by NSS at 15,848 Mbps, which is lower than the vendor-claimed performance; Fortinet rates this device at 23,000 Mbps. *NSS-Tested Throughput* is calculated as a weighted average of the traffic that NSS expects an NGIPS to experience in an enterprise environment. For more details, please see Appendix A: Product Scorecard.

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

[2] In accordance with the industry standard for vulnerability disclosures and to provide vendors with sufficient time to add protection where necessary, NSS Labs will not publicly release information about which previously unpublished techniques were applied during testing until 90 days after the publication of this document.

Table of Contents

Table of Figures

Security Effectiveness

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

False Positive Testing

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

NSS Exploit Library

NSS' security effectiveness testing leverages the deep expertise of our engineers who utilize multiple commercial, open-source, and proprietary tools as appropriate. With more than 1,900 exploits, this is the industry's most comprehensive test to date.

Product	Total Number of Attacks Run	Total Number of Attacks Blocked	Block Percentage
Fortinet FortiGate 3000D v5.6.4GA build 7892	2,073	2,060	99.4%

Figure 2 – Number of Attacks Blocked (%)

Resiliency

NSS also measured the resiliency of a device by introducing previously unseen variations of a known exploit and measuring the device's effectiveness against them. Figure 3 depicts the resiliency score.

Product	Block Percentage
Fortinet FortiGate 3000D v5.6.4GA build 7892	86.2%

Figure 3 –Resiliency Score

Coverage by Impact Type

The most serious exploits are those that result in a remote system compromise, providing the attacker with the ability to execute arbitrary system-level commands. Most exploits in this class are "weaponized" and offer the attacker a fully interactive remote shell on the target client or server. Slightly less serious are attacks that result in individual service compromise but not arbitrary system-level command execution, but this distinction is becoming less relevant in the modern threat landscape. Finally, there are attacks that result in a system- or service-level fault that crashes the targeted service or application and requires administrative action to restart the service or reboot the system. Clients can contact NSS for more information about these tests.

Coverage by Date

Figure 4 provides insight into whether or not a vendor is aging out protection signatures aggressively enough to preserve performance levels. It also reveals whether a product lags behind in protection for the most current vulnerabilities. NSS reports exploits by individual years for the past ten years. Exploits older than ten years are grouped together.

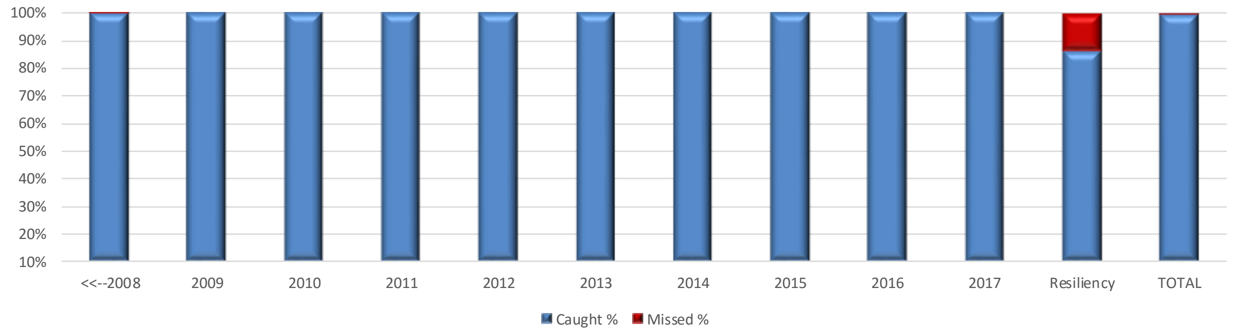


Figure 4 – Product Coverage by Date

Coverage by Target Vendor

Exploits within the *NSS Exploit Library* target a wide range of protocols and applications. Figure 5 depicts the coverage offered by the FortiGate 3000D for five of the top vendors targeted in this test. More than 70 vendors are represented in the test. Clients can contact NSS for more information.

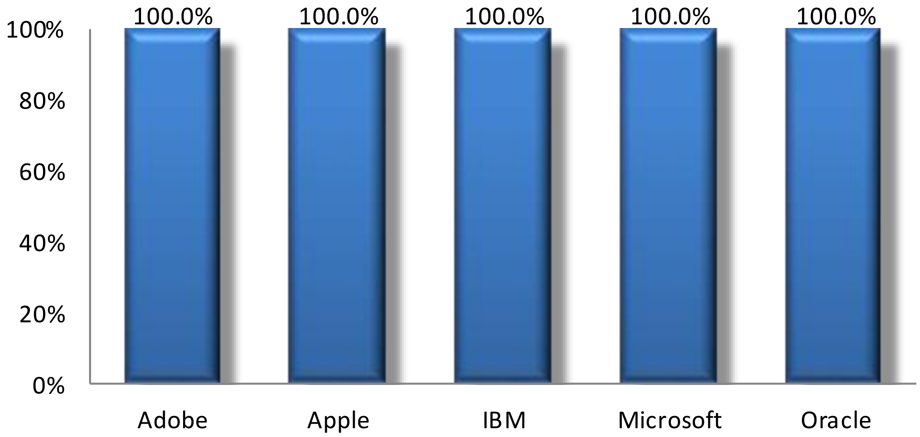


Figure 5 – Product Coverage by Target Vendor

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 Exploits

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

Protection from web-based exploits targeting client applications, also known as "drive-by" downloads, can be effectively measured in NSS' unique live test harness through a series of procedures that measure the stages of protection. Unlike traditional malware that is downloaded and installed, "drive-by" attacks first exploit a vulnerable application then silently download and install malware. For more information, see the Comparative Report on Security.

Product	Block Percentage
Fortinet FortiGate 3000D v5.6.4GA build 7892	100.0%

Figure 6 – Number of Attacks Blocked (%)

Resistance to Evasion Techniques

Evasion techniques are a means of disguising and modifying attacks at the point of delivery to avoid detection and blocking by security products. Failure by a security device to correctly identify a specific type of evasion potentially allows an attacker to use an entire class of exploits for which the device is assumed to have protection. This often renders the device virtually useless. Many of the techniques used in this test have been widely known for years and should be considered minimum requirements for the NGIPS product category.

Providing exploit protection results without fully factoring in evasions can be misleading. The more classes of evasion that are missed (such as HTTP evasions, IP packet fragmentation, TCP stream segmentation, HTML obfuscation and resiliency), the less effective the device. For example, it is better to miss all techniques in one evasion category, such as IP packet fragmentation, than one technique in each category, which would result in a broader attack surface.

Furthermore, evasions operating at the lower layers of the network stack (IP packet fragmentation or stream segmentation) have a greater impact on security effectiveness than those operating at the upper layers (HTTP evasions or HTML obfuscation.) Lower-level evasions will potentially impact a wider number of exploits; missing TCP segmentation, for example, is a much more serious issue than missing FTP obfuscation.

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

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

The resiliency of a system can be defined as its ability to absorb an attack and reorganize around a threat. When an attacker is presented with a vulnerability, the attacker can select one or more paths to trigger the vulnerability. NSS will measure a device's resiliency by introducing a vulnerability along with its triggers and then asking the device to protect against the vulnerability. NSS will introduce various, previously unseen variations of exploits to exploit the vulnerability and measure the device's effectiveness against them. A resilient device will be able to detect and prevent against different variations of the exploit. For more, see the Evasions Test Methodology v1.1 at www.nsslabs.com. Figure 7 provides the results of the evasion tests for the FortiGate 3000D.

Test Procedure	Result
HTML Evasions	PASS
IP Packet Fragmentation + TCP Segmentation	PASS
HTTP Evasions	PASS
TCP Split Handshake	PASS
Resiliency[4]	
Attacks on nonstandard ports[5]	PASS

Figure 7 – Resistance to Evasion Results

[4] The results of resiliency testing are included in the Exploit Block Rate calculations.

[5] Enterprises should be aware of the importance of egress filtering and should ensure their configurations mitigate these risks.

Performance

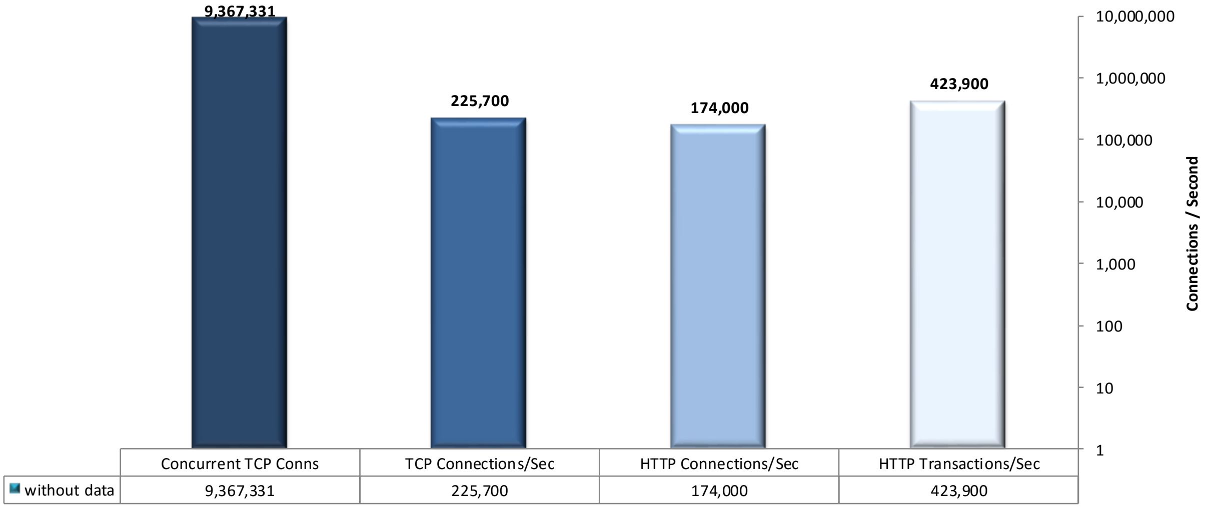
There is frequently a trade-off between security effectiveness and performance. Because of this trade-off, it is important to judge a product's security effectiveness within the context of its performance and vice versa. This ensures that new security protections do not adversely impact performance and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

Maximum Capacity

The use of traffic generation appliances allows NSS engineers to create "real-world" traffic at multi-Gigabit speeds as a background load for the tests. The aim of these tests is to stress the inspection engine and determine how it copes with high volumes of TCP connections per second, application-layer transactions per second, and concurrent open connections. All packets contain valid payload and address data, and these tests provide an excellent representation of a live network at various connection/transaction rates.

Note that in all tests the following critical "breaking points"—where the final measurements are taken—are used:

- **Excessive concurrent TCP connections** – Latency within the NGIPS is causing an unacceptable increase in open connections.
- **Excessive concurrent HTTP connections** – Latency within the NGIPS is causing excessive delays and increased response time.
- **Unsuccessful HTTP transactions** – Normally, there should be zero unsuccessful transactions. Once these appear, it is an indication that excessive latency within the NGIPS is causing connections to time out.



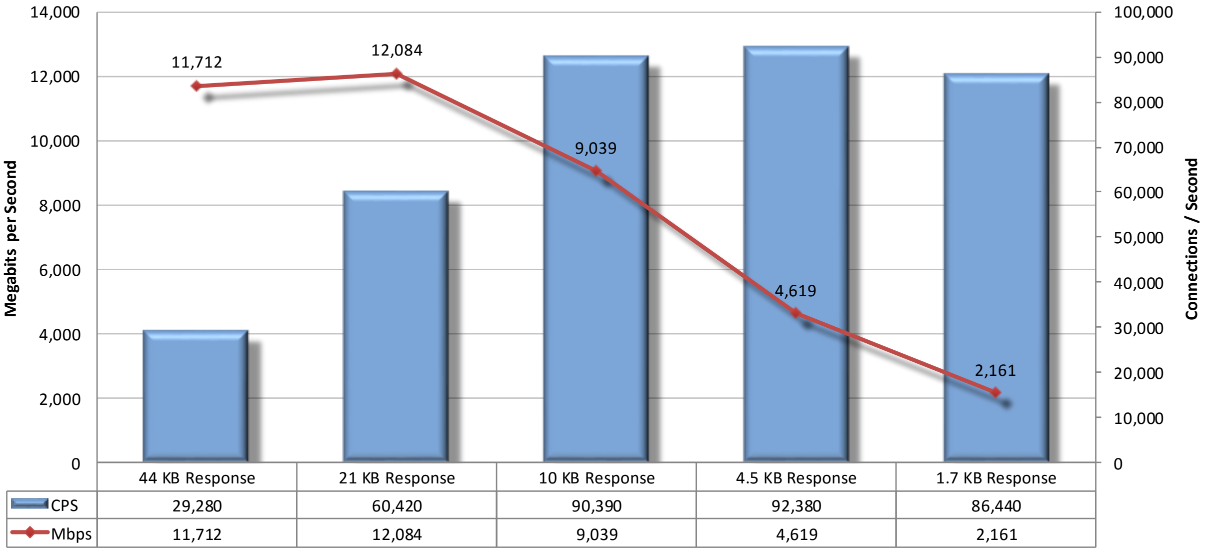
	Concurrent TCP Conns	TCP Connections/Sec	HTTP Connections/Sec	HTTP Transactions/Sec
without data	9,367,331	225,700	174,000	423,900

Figure 8 – Concurrency and Connection Rates

HTTP Capacity

The aim of the HTTP capacity tests is to stress the HTTP detection engine and determine how the device copes with network loads of varying average packet size and varying connections per second. By creating multiple tests using genuine session-based traffic with varying session lengths, the device is forced to track valid HTTP sessions, thus ensuring a higher workload than for simple packet-based background traffic.

Each transaction consists of a single HTTP GET request. All packets contain valid payload (a mix of binary and ASCII objects) and address data. This test provides an excellent representation of a live network (albeit one biased toward HTTP traffic) at various network loads.



	44 KB Response	21 KB Response	10 KB Response	4.5 KB Response	1.7 KB Response
CPS	29,280	60,420	90,390	92,380	86,440
Mbps	11,712	12,084	9,039	4,619	2,161

Figure 9 – HTTP Capacity

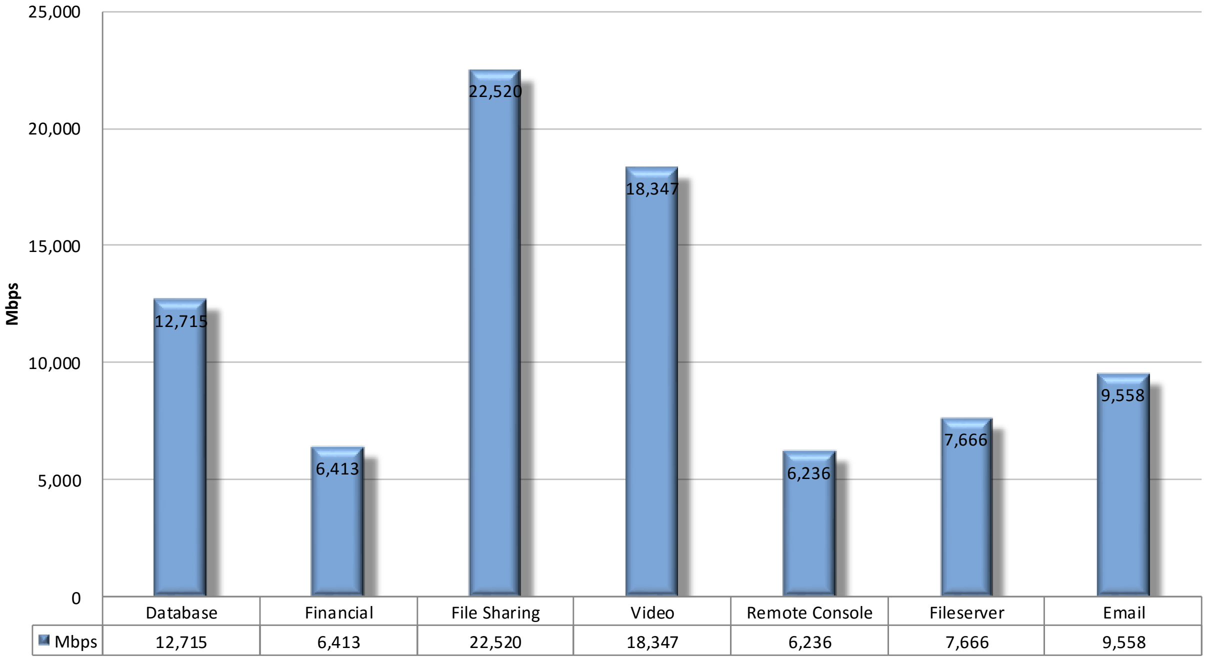
Application Average Response Time – HTTP

Application Average Response Time – HTTP (at 90% Maximum Load)	Milliseconds
2,500 Connections per Second – 44 KB Response	0.35
5,000 Connections per Second – 21 KB Response	0.55
10,000 Connections per Second – 10 KB Response	0.78
20,000 Connections per Second – 4.5 KB Response	0.59
40,000 Connections per Second – 1.7 KB Response	0.37

Figure 10 – Average Application Response Time (Milliseconds)

Single Application Flows

This test measures the performance of the device with single application flows. For details about single application flow testing, see the NSS Labs Next Generation Intrusion Prevention System (NGIPS) Test Methodology v4, available at www.nsslabs.com.



	Database	Financial	File Sharing	Video	Remote Console	Fileserver	Email
▪ Mbps	12,715	6,413	22,520	18,347	6,236	7,666	9,558

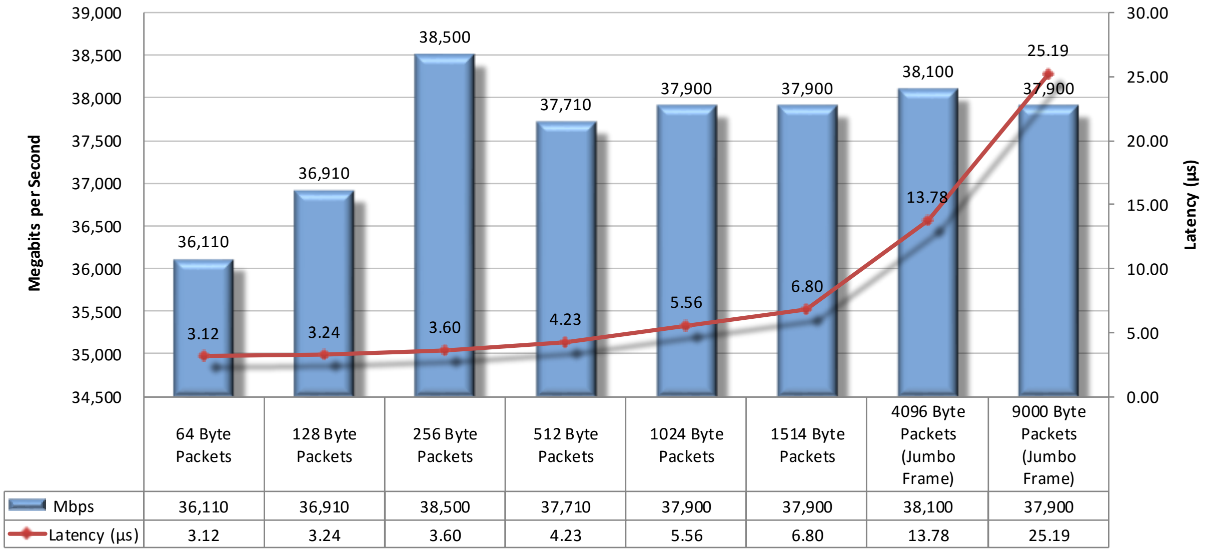
Figure 11 – Single Application Flows

Raw Packet Processing Performance (UDP Throughput)

This test uses UDP packets of varying sizes generated by test equipment. A constant stream of the appropriate packet size along with variable source and destination IP addresses is transmitted bidirectionally through each port pair of the device.

Each packet contains dummy data and is targeted at a valid port on a valid IP address on the target subnet. The percentage load and frames per second (fps) figures across each inline port pair are verified by network monitoring tools before each test begins. Multiple tests are run and averages are taken where necessary.

This traffic does not attempt to simulate any real-world network condition. The aim of the test is to determine the raw packet processing capability of each inline port pair of the device as well as the device's effectiveness at forwarding packets quickly in order to provide the highest level of network performance with the least amount of latency.



	64 Byte Packets	128 Byte Packets	256 Byte Packets	512 Byte Packets	1024 Byte Packets	1514 Byte Packets	4096 Byte Packets (Jumbo Frame)	9000 Byte Packets (Jumbo Frame)
Mbps	36,110	36,910	38,500	37,710	37,900	37,900	38,100	37,900
Latency (µs)	3.12	3.24	3.60	4.23	5.56	6.80	13.78	25.19

Figure 12 – Raw Packet Processing Performance (UDP Traffic)

Raw Packet Processing Performance (UDP Latency)

NGIPS that introduce high levels of latency lead to unacceptable response times for users, especially where multiple security devices are placed in the data path. Figure 13 depicts UDP latency (in microseconds) as recorded during the UDP throughput tests at 90% of maximum load.

Latency – UDP	Microseconds
64-Byte Packets	3.12
128-Byte Packets	3.24
256-Byte Packets	3.60
512-Byte Packets	4.23
1024-Byte Packets	5.56
1514-Byte Packets	6.80
4096 Byte Packets (Jumbo Frame)	13.78
9000 Byte Packets (Jumbo Frame)	25.19

Figure 13 – UDP Latency in Microseconds

Stability and Reliability

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

The device is required to remain operational and stable throughout all these tests, and to block 100% of previously known malicious attacks, raising an alert for each. If any non-allowed traffic passes successfully, caused either by the volume of traffic or by the device failing open for any reason, it will fail the test.

Stability and Reliability	Result
Blocking under Extended Attack	PASS
Passing Legitimate Traffic under Extended Attack	PASS
Power Fail Recovery	PASS
Power Redundancy	YES
Power Fail Open (No Inspection) [6]	NO
Persistence of Data	PASS

Figure 14 – Stability and Reliability Results

[6] Not included in the stability and reliability score; included only to serve as a reference point.

Total Cost of Ownership (TCO)

Implementation of security products can be complex, with several factors affecting the overall cost of deployment, maintenance, and upkeep. Each of the following should be considered over the course of the useful life of the product:

- **Product Purchase** – The cost of acquisition.
- **Product Maintenance** – The fees paid to the vendor, including software and hardware support, maintenance, and other updates.
- **Installation** – The time required to take the device out of the box, configure it, put it into the network, apply updates and patches, and set up desired logging and reporting.
- **Upkeep** – The time required to apply periodic updates and patches from vendors, including hardware, software, and other updates.
- **Management** – Day-to-day management tasks, including device configuration, policy updates, policy deployment, alert handling, and so on.

For the purposes of this report, capital expenditure (capex) items are included for a single device only (the cost of acquisition and installation).

Installation Hours

Figure 15 depicts the number of hours of labor required to install each device using only local device management options. The table accurately reflects the amount of time that NSS engineers, with the help of vendor engineers, needed to install and configure the device to the point where it operated successfully in the test harness, passed legitimate traffic, and blocked and detected prohibited or malicious traffic. This closely mimics a typical enterprise deployment scenario for a single device.

The installation cost is based on the time that an experienced security engineer would require to perform the installation tasks described above. This approach allows NSS to hold constant the talent cost and measure only the difference in time required for installation. Readers should substitute their own costs to obtain accurate TCO figures.

Product	Installation (Hours)
Fortinet FortiGate 3000D v5.6.4GA build 7892	8

Figure 15 – Sensor Installation Time (Hours)

Total Cost of Ownership

Calculations are based on vendor-provided pricing information. Where possible, the 24/7 maintenance and support option with 24-hour replacement is utilized, since this is the option typically selected by enterprise customers. Prices are for single device management and maintenance only; costs for central management solutions (CMS) may be extra.

Product	Year 1 Cost	Year 2 Cost	Year 3 Cost	3-Year TCO
Fortinet FortiGate 3000D v5.6.4GA build 7892	$54,975	$16,875	$16,875	$88,725

Figure 16 –3-Year TCO (US$)

- **Year 1 Cost** is calculated by adding installation costs (US$75 per hour fully loaded labor x installation time) + purchase price + first-year maintenance/support fees.
- **Year 2 Cost** consists only of maintenance/support fees.
- **Year 3 Cost** consists only of maintenance/support fees.

For additional TCO analysis, including for the CMS, refer to the TCO Comparative Report.

Appendix A: Product Scorecard

Security Effectiveness	
False Positive Testing	PASS
NSS Labs Exploit Library	99.4%
Live Drive-By Exploits	100.0%
Combined Block Rate	99.6%
Resistance to Evasion	
TCP Split Handshake	PASS
HTTP Evasions	
http-0.9-001 (HTTP/0.9 response (no response headers))	PASS
http-0.9-002 (Declared HTTP/0.9 response; but includes response headers; space (hex '20') after server header)	PASS
http-ch-001 (HTTP/1.1 chunked response with chunk sizes followed by a space (hex '20'))	PASS
http-ch-002 (HTTP/1.1 chunked response with chunk sizes followed by a tab (hex '09'))	PASS
http-ch-003 (HTTP/1.1 chunked response with chunk sizes followed by an 'x' (hex '78'))	PASS
http-ch-004 (HTTP/1.1 chunked response with chunk sizes followed by a comma (hex '2c'))	PASS
http-ch-005 (HTTP/1.1 chunked response with chunk sizes followed by null character (hex '00'))	PASS
http-ch-006 (HTTP/1.1 chunked response with chunk sizes followed by a vertical tab (hex '0b'))	PASS
http-ch-007 (HTTP/1.1 chunked response with chunk sizes followed by form feed (hex '0c'))	PASS
http-ch-008 (HTTP/1.1 chunked response with final chunk size of '00' (hex '20 20' rather than hex '20'))	PASS
http-ch-009 (HTTP/1.1 chunked response with final chunk size of '00000000000000000000' (rather than '0'))	PASS
http-ch-010 (HTTP/1.1 chunked response with chunk sizes followed by a space (hex '20') then an 'x' (hex '78'))	PASS
http-ch-011 (HTTP/1.1 response with line folded transfer-encoding header declaring chunking ('Transfer-Encoding: ' followed by CRLF (hex '0d 0a') followed by space (hex '20') followed by 'chunked' followed by CRLF (hex '0d 0a')); served without chunking)	PASS
http-ch-012 (HTTP/1.1 response with transfer-encoding header declaring chunking with lots of whitespace ('Transfer-Encoding: ' followed by 500 spaces (hex '20' * 500) followed by 'chunked' followed by CRLF (hex '0d 0a')); served chunked)	PASS
http-ch-013 (HTTP/1.0 response declaring chunking; served without chunking)	PASS
http-ch-014 (HTTP/1.0 response declaring chunking with content-length header; served without chunking)	PASS
http-ch-015 (<tab>Transfer-Encoding: chunked as first header line; served chunked)	PASS
http-ch-016 (<tab>Transfer-Encoding: chunked as continuation of some header line; served chunked)	PASS
http-ch-017 (header with no field name and the string ": open"; followed by header beginning with a tab (hex '09') followed by 'Transfer-Encoding: chunked'; followed by header 'Custom-header: check'; served chunked)	PASS
http-ch-018 (TE chunked prefixed with <CR><CR>;served chunked)	PASS
http-ch-019 (HTTP/1.1\nTransfer-Encoding:chunked; served chunked)	PASS
http-ch-020 (HTTP/1.1\r\nTransfer-Encoding:chunked; served chunked)	PASS
http-ch-021 (single \n instead of \r\n and chunked)	PASS
http-ch-022 (HTTP/1.1\rTransfer-Encoding:chunked; served chunked)	PASS
http-ch-023 (double <LF> before header; chunked)	PASS
http-ch-024 (double <CR><LF> before header; chunked)	PASS
http-ch-025 (junk followed by single <CR><LF> before header; chunked)	PASS
http-ch-026 (SIP/2.0 200 ok followed by single <CR><LF> before header; chunked)	PASS
http-ch-027 (space+junk followed by single <CR><LF> before header; chunked)	PASS
http-ch-028 (space+"SIP/2.0 200 ok" followed by single <CR><LF> before header; chunked)	PASS
http-ch-029 (single <LF> before header; chunked)	PASS
http-ch-030 (H before header; chunked)	PASS
http-ch-031 (HT before header; chunked)	PASS

http-ch-032 (HTT before header; chunked)	PASS
http-ch-033 (HTTX before header; chunked)	PASS
http-ch-034 (HTTXY before header; chunked)	PASS
http-gz-001 (HTTP/1.1 response with content-encoding header for gzip; followed by content-encoding header for deflate; no space between ':' and declaration of encoding types; served with no compression)	PASS
http-gz-002 (HTTP/1.1 response with content-encoding declaration of "gzip x"; served uncompressed)	PASS
http-gz-003 (header end \n\r\n; gzip)	PASS
http-gz-004 (header end \n\r\n; gzip with content-length)	PASS
http-gz-005 (header end \n\013\n\n and gzip)	PASS
http-gz-006 (header end \n\013\n\n and gzip with content length)	PASS
http-gz-007 (header end \r\n\013\r\n\r\n and gzip)	PASS
http-gz-008 (header end \r\n\013\r\n\r\n and gzip with content-length)	PASS
http-gz-009 (header end \n\r\r\n; gzip)	PASS
http-gz-010 (header end \n\r\r\n; gzip with content-length)	PASS
http-gz-011 (header end "\n\x20 \n" and gzip)	PASS
http-gz-012 (header end "\n\x20\n" and gzip with content-length)	PASS
http-gz-013 (header end \n\011\n and gzip)	PASS
http-gz-014 (header end \n\011\n and gzip with content-length)	PASS
http-gz-015 (header end \n\n; gzip)	PASS
http-stat-001 (HTTP/1.0 response with status code 100 followed by message-body; no content-length header)	PASS
http-stat-002 (HTTP/1.0 response with status code 206 followed by message-body; no content-length header)	PASS
http-stat-003 (HTTP/1.0 response with status code 304 followed by message-body; no content-length header)	PASS
http-stat-004 (HTTP/1.0 response with status code 404 followed by message-body; no content-length header)	PASS
http-stat-005 (HTTP/1.0 response with status code 500 followed by message-body; no content-length header)	PASS
http-stat-006 (HTTP/1.0 response with status code 600 followed by message-body; no content-length header)	PASS
http-stat-007 (HTTP/1.0 response with status code 900 followed by message-body; no content-length header)	PASS
http-stat-008 (status code 101 with body)	PASS
http-stat-009 (status code 102 with body)	PASS
http-cmb-001 (HTTP/1.1 response with content-length header size declaration followed by space and letter A(hex'2041');message-body followed by junk (e.g. '</html>HBGIBFJ236MJXICVNGRXKRADDPXAMVOLLCCK3KXWGBOP0TKBNKQEGS7MM0EOEHDTDZIY553OGE'))	PASS
http-cmbch-002 (Chunked Header and HTTP/1.01. Served chunked)	PASS
http-cmbch-003 (Chunked Header and HTTP/1.10. Served chunked)	PASS
http-cmbcg-004 (Chunked Header and HTTP/01.1. Served chunked and with gzip)	PASS
http-cmbcg-005 (Chunked Header and HTTP/11.01. Served chunked and with gzip)	PASS
http-cmbcg-006 (Chunked Header and HTTP/11.10. Served chunked and with gzip)	PASS
http-cmbch-008 (version HTTP/1.010 instead of HTTP/1.1 and chunked)	PASS
http-cmbch-009 (version HTTP/2.B instead of HTTP/1.1 and chunked)	PASS
http-cmbch-010 (version HTTP/9.-1 instead of HTTP/1.1 and chunked)	PASS
http-cmbgz-011 (double Transfer-Encoding: first empty; last chunked. Served with content-length and gzipped; not chunked)	PASS
IP Packet Fragmentation/ TCP Segmentation	
net-ip-001 (overlapping small IP fragments favoring new data)	PASS
net-ip-002 (overlapping small IP fragments favoring new data in reverse order)	PASS
net-ip-003 (overlapping small IP fragments favoring new data in random order)	PASS
net-ip-006 (overlapping small IP fragments favoring new data; interleave chaff (invalid IP options))	PASS
net-ip-007 (overlapping small IP fragments favoring new data in random order; interleave chaff (invalid IP options))	PASS

net-ip-008 (overlapping small IP fragments favoring new data in random order; interleave chaff (invalid IP options); delay random fragment)	PASS
net-ip-009 (overlapping small IP fragments favoring new data; interleave chaff (invalid IP options); DSCP value 16)	PASS
net-ip-010 (overlapping small IP fragments favoring new data in random order; interleave chaff (invalid IP options); delay random fragment; DSCP value 34)	PASS
net-ip-011 (small IP fragments)	PASS
net-ip-012 (small IP fragments in reverse order)	PASS
net-ip-013 (small IP fragments in random order)	PASS
net-ip-014 (small IP fragments; delay first fragment)	PASS
net-ip-015 (small IP fragments in reverse order; delay last fragment)	PASS
net-ip-016 (small IP fragments; interleave chaff (invalid IP options))	PASS
net-ip-017 (small IP fragments in random order; interleave chaff (invalid IP options))	PASS
net-ip-018 (small IP fragments in random order; interleave chaff (invalid IP options); delay random fragment)	PASS
net-ip-019 (small IP fragments; interleave chaff (invalid IP options); DSCP value 16)	PASS
net-ip-020 (small IP fragments in random order; interleave chaff (invalid IP options); delay random fragment; DSCP value 34)	PASS
net-tcp-001 (overlapping small TCP segments favoring new data)	PASS
net-tcp-002 (overlapping small TCP segments favoring new data in reverse order)	PASS
net-tcp-003 (overlapping small TCP segments favoring new data in random order)	PASS
net-tcp-004 (overlapping small TCP segments favoring new data; delay first segment)	PASS
net-tcp-005 (overlapping small TCP segments favoring new data in reverse order; delay last segment)	PASS
net-tcp-006 (overlapping small TCP segments favoring new data; interleave chaff (invalid TCP checksums); delay first segment)	PASS
net-tcp-007 (overlapping small TCP segments favoring new data in random order; interleave chaff (older PAWS timestamps); delay last segment)	PASS
net-tcp-008 (overlapping small TCP segments favoring new data in random order; interleave chaff (out-of-window sequence numbers); TCP MSS option)	PASS
net-tcp-009 (overlapping small TCP segments favoring new data in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option)	PASS
net-tcp-010 (overlapping small TCP segments favoring new data in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option; delay first segment)	PASS
net-tcp-011 (small TCP segments)	PASS
net-tcp-012 (small TCP segments in reverse order)	PASS
net-tcp-013 (small TCP segments in random order)	PASS
net-tcp-014 (small TCP segments; delay first segment)	PASS
net-tcp-015 (small TCP segments in reverse order; delay last segment)	PASS
net-tcp-016 (small TCP segments; interleave chaff (invalid TCP checksums); delay first segment)	PASS
net-tcp-017 (small TCP segments in random order; interleave chaff (older PAWS timestamps); delay last segment)	PASS
net-tcp-018 (small TCP segments in random order; interleave chaff (out-of-window sequence numbers); TCP MSS option)	PASS
net-tcp-019 (small TCP segments in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option)	PASS
net-tcp-020 (small TCP segments in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option; delay first segment)	PASS
net-cmb-001 (overlapping small TCP segments favoring new data; small IP fragments, Listener port 80)	PASS
net-cmb-002 (small TCP segments; overlapping small IP fragments favoring new data, Listener port 80)	PASS
net-cmb-003 (overlapping small TCP segments favoring new data; overlapping small IP fragments favoring new data)	PASS
net-cmb-004 (overlapping small TCP segments favoring new data in random order; small IP fragments in random order)	PASS

net-cmb-005 (small TCP segments in random order; overlapping small IP fragments favoring new data in random order)	PASS
net-cmb-006 (overlapping small TCP segments favoring new data in random order; overlapping small IP fragments favoring new data in random order)	PASS
net-cmb-007 (overlapping small TCP segments favoring new data in random order; overlapping small IP fragments favoring new data in random order; interleave chaff (invalid IP options))	PASS
net-cmb-008 (overlapping small TCP segments favoring new data; interleave chaff (invalid TCP checksums); small IP fragments; interleave chaff (invalid IP options))	PASS
net-cmb-009 (small TCP segments; interleave chaff (invalid TCP checksums); overlapping small IP fragments favoring new data; interleave chaff (invalid IP options))	PASS
net-cmb-010 (small TCP segments; interleave chaff (invalid TCP checksums); delay last segment; overlapping small IP fragments favoring new data; interleave chaff (invalid IP options))	PASS
net-cmb-011 (small TCP segments; small IP fragments)	PASS
net-cmb-012 (small TCP segments; small IP fragments in reverse order)	PASS
net-cmb-013 (small TCP segments in random order; small IP fragments)	PASS
net-cmb-014 (small TCP segments; small IP fragments in random order)	PASS
net-cmb-015 (small TCP segments in random order; small IP fragments in reverse order)	PASS
net-cmb-016 (small TCP segments in random order; interleave chaff (invalid TCP checksums); small IP fragments in reverse order; interleave chaff (invalid IP options))	PASS
net-cmb-017 (small TCP segments; interleave chaff (invalid TCP checksums); delay last segment; small IP fragments; interleave chaff (invalid IP options))	PASS
net-cmb-018 (small TCP segments; interleave chaff (invalid TCP checksums); small IP fragments; interleave chaff (invalid IP options); delay last fragment)	PASS
net-cmb-019 (small TCP segments in random order; interleave chaff (out-of-window sequence numbers); TCP MSS option; small IP fragments in random order; interleave chaff (invalid IP options); delay random fragment)	PASS
net-cmb-020 (small TCP segments in random order; interleave chaff (requests to resynch sequence numbers mid-stream); TCP window scale option; delay first segment; small IP fragments)	PASS
HTML Evasions	
html-utf-001 (UTF-8 encoding)	PASS
html-utf-002 (UTF-8 encoding with BOM)	PASS
html-utf-003 (UTF-16 encoding)	PASS
html-utf-004 (UTF-8 encoding; no http or html declarations)	PASS
html-utf-005 (UTF-8 encoding with BOM; no http or html declarations)	PASS
html-utf-006 (UTF-16 encoding with BOM; no http or html declarations)	PASS
html-pad-001 (padded with 1MB)	PASS
html-pad-002 (padded with 15MB)	PASS
html-pad-003 (padded with 30MB)	PASS
html-padch-001 (padded with 1MB and chunked)	PASS
html-padch-002 (padded with 15MB and chunked)	PASS
html-padch-003 (padded with 30MB and chunked)	PASS
html-padgz-001 (padded with 1MB and compressed with gzip)	PASS
html-padgz-002 (padded with 15MB and compressed with gzip)	PASS
html-padgz-003 (padded with 30MB and compressed with gzip)	PASS
html-padde-001 (padded with 1MB and compressed with deflate)	PASS
html-padde-002 (padded with 15MB and compressed with deflate)	PASS
html-padde-003 (padded with 30MB and compressed with deflate)	PASS
Resiliency	
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS

Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS

Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	PASS
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	FAIL
Information withheld for 90 days. See Footnote 2	FAIL
Attacks on nonstandard ports	PASS

Performance		
Raw Packet Processing Performance (UDP Traffic)	NSS-Rated Throughput (Mbps) Weighting	Mbps
64 Byte Packets	0%	36,110
128 Byte Packets	1%	36,910
256 Byte Packets	1%	38,500
512 Byte Packets	1%	37,710
1024 Byte Packets	3%	37,900
1514 Byte Packets	3%	37,900
4096 Byte Packets	3%	38,100
9000 Byte Packets	3%	37,900
Latency – UDP		Microseconds
64 Byte Packets		3.12
128 Byte Packets		3.24
256 Byte Packets		3.60
512 Byte Packets		4.23
1024 Byte Packets		5.56
1514 Byte Packets		6.80
4096 Byte Packets		13.78
9000 Byte Packets		25.19
Maximum Capacity		CPS
Theoretical Max. Concurrent TCP Connections		9,367,331
Maximum TCP Connections Per Second		225,700
Maximum HTTP Connections Per Second		174,000
Maximum HTTP Transactions Per Second		423,900
HTTP Capacity with No Transaction Delays	NSS-Rated Throughput (Mbps) Weighting	CPS
2,500 Connections Per Second – 44Kbyte Response	8%	29,280
5,000 Connections Per Second – 21Kbyte Response	8%	60,420
10,000 Connections Per Second – 10Kbyte Response	7%	90,390
20,000 Connections Per Second – 4.5Kbyte Response	7%	92,380
40,000 Connections Per Second – 1.7Kbyte Response	4%	86,440
Application Average Response Time – HTTP (at 90% Max Load)		Milliseconds
2.500 Connections Per Second – 44Kbyte Response		0.3484
5,000 Connections Per Second – 21Kbyte Response		0.5462
10,000 Connections Per Second – 10Kbyte Response		0.7823
20,000 Connections Per Second – 4.5Kbyte Response		0.5881
40,000 Connections Per Second – 1.7Kbyte Response		0.3658
“Real-World” Single App Traffic	NSS-Rated Throughput (Mbps) Weighting	Mbps
Database	6%	12,715
Financial	0%	6,413
File Sharing	10%	22,520
Video	14%	18,347
Remote Console	1%	6,236
Fileserver	8%	7,666
Email	13%	9,558
Stability & Reliability		
Blocking Under Extended Attack		PASS
Passing Legitimate Traffic Under Extended Attack		PASS
Power Fail Recovery		PASS

Power Redundancy	YES
Power Fail Open (No Inspection)	NO
Persistence of Data	PASS
Total Cost of Ownership	
Ease of Use	
Initial Setup (Hours)	8
Time Required for Upkeep (Hours per Year)	Contact NSS Labs
Time Required to Tune (Hours per Year)	Contact NSS Labs
Expected Costs	
Initial Purchase (hardware as tested)	$37,500
Installation Labor Cost (@$75/hr)	$600
Annual Cost of Maintenance & Support (hardware/software)	$16,875
Annual Cost of Updates (IPS/AV/etc.)	$0
Initial Purchase (centralized management system)	Contact NSS Labs
Annual Cost of Maintenance & Support (centralized management system)	Contact NSS Labs
Management Labor Cost (per Year @$75/hr)	Contact NSS Labs
Tuning Labor Cost (per Year @$75/hr)	Contact NSS Labs
Total Cost of Ownership	
Year 1	$54,975
Year 2	$16,875
Year 3	$16,875
3 Year Total Cost of Ownership	$88,725

Figure 17 – Detailed Scorecard

Test Methodology

NSS Labs Next Generation Intrusion Prevention System (NGIPS) Test Methodology v4.0

NSS Labs Evasions Test Methodology v1.1

Contact Information

3711 South Mopac Expressway
Building 1, Suite 400
Austin, TX 78746
info@nsslabs.com
www.nsslabs.com

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

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.