NEXT GENERATION FIREWALL PRODUCT ANALYSIS

Sourcefire 8250

2012

1 Introduction

Over the past decade, complex business processes have changed the way firewalls are used and accessed. As a consequence of this, the security landscape has changed significantly, with Web 2.0 trends pushing critical business applications through firewall ports that were previously reserved for a single function, such as HTTP. The effect is that the legacy firewall technology is effectively blinded, and unable to differentiate between actual HTTP traffic and non-HTTP services tunneling over port 80 (such as VoIP or instant messaging). The security administrator is powerless to stop this without crippling essential business processes.

This means that relying simply on IP address and port combinations to define network applications is no longer sufficient. Firewalls need to be capable of performing deep packet inspection on all packets, ports, and protocols in order to determine which applications are running on the network.

NSS Labs' research also indicates that over the past 18 months, the sophistication and strategic capabilities of cybercriminals has outstripped the rate of evolution of information security products. In addition to traditional remote attacks against servers, cybercriminals are increasingly waging highly targeted campaigns against desktop client applications.

Simply enforcing proper and compliant protocol use on standard ports and preventing attacks against unpatched servers are no longer of sufficient value in this environment. To meet these challenges, firewalls need to evolve into "next-generation" firewalls (NGFW). These "next-generation" firewalls combine legacy firewall capabilities with Intrusion prevention systems (IPS) and incorporate advanced application and user ID awareness to enable the creation of granular security policies capable of operating in a Web 2.0 world.

The following capabilities are considered essential as part of a NGFW device:

- Traditional "first generation firewall" including:
  - Basic packet filtering
  - Stateful multi-layer inspection
  - NAT
  - VPN
  - Highly Stable
  - High Availability
- Integrated IPS
- Application awareness/control

- User/group control
- Ability to operate at layer 3 ("traditional") or layer 2 ("bump in the wire")

The test results below are based upon NSS Labs' *NGFW Test Methodology v4.0*. For additional information on NGFW technology, please refer to the NSS Labs Analysis Brief entitled "*What Do CIOs Need to Know About Next Generation Firewalls?*"

Table of Contents

1 Introduction 1

2 Summary Results 5

2.1 Firewall Policy Enforcement 6

2.2 Application Control 6

2.3 User/Group ID Aware Policies 7

2.4 Intrusion Prevention 8

2.4.1 Coverage by Attack Vector 8

2.4.2 Coverage by Impact Type 9

2.4.3 Coverage by Date 10

2.4.4 Coverage by Target Vendor 10

2.4.5 Coverage by Result 11

2.4.6 Coverage by Target Type 11

2.4.7 Attack Leakage 11

2.5 Resistance to Evasion 12

3 Performance 12

3.1 Connection Dynamics – Concurrency and Connection Rates 12

3.2 HTTP Connections per Second and Capacity 13

3.3 Application Average Response Time – HTTP 14

3.4 HTTP Connections per Second and Capacity (With Delays) 14

3.5 UDP Throughput 15

3.6 Latency – UDP 16

3.7 Real-World Traffic Mixes 16

4 Stability & Reliability 17

5 Management & Configuration 18

5.1 General 18

5.2 Policy 19

5.3 Alert Handling 20

5.4 Reporting 22

6 Total Cost of Ownership (TCO) 22

6.1 Labor per Product (in Hours) 23

6.2 Purchase Price and Total Cost of Ownership 23

6.3 Value: Cost per Mbps and Exploit Blocked 23

7 Detailed Product Scorecard 24

Contact Information 31

Table of Contents

Figure 1: Coverage by Attack Vector 9

Figure 2: Product Coverage by Impact 10

Figure 3: Product Coverage by Date 10

Figure 4: Product Coverage by Target Vendor 11

Figure 5: Concurrency and Connection Rates 13

Figure 6: HTTP Connections per Second and Capacity 14

Figure 7: HTTP Connections per Second and Capacity (With Delays) 15

Figure 8: UDP Throughput 16

Figure 9: Real-World Traffic Mixes 17

2 Summary Results

NSS Labs performed an independent test of the Sourcefire 8250 5.1 NGFW. The product was subjected to thorough testing at the NSS Labs facility in Austin, Texas, based on the NGFW methodology v4.0 available on www.nsslabs.com. This test was conducted free of charge and NSS Labs did not receive any compensation in return for Sourcefire's participation.

While the NGFW Comparative Analysis Reports (CAR) on security, performance, management, and total cost of ownership (TCO) will provide high-level comparative data on all tested products, this in-depth Product Analysis Report (PAR) provides detailed information not available elsewhere.

NSS research indicates that the majority of enterprises do not tune the IPS module separately within their NGFW. Therefore, NSS Labs' evaluation of NGFW products is configured with the vendor pre-defined or default, "out-of-the-box" settings, in order to provide readers with relevant security effectiveness and performance dimensions based upon their expected usage.

As part of this test, **Sourcefire** submitted the **8250 5.1.**

Product	Overall Protection	Client Protection	Throughput
Sourcefire 8250 5.1	98.9%	99.1%	10,000 Mbps
Stability & Reliability	Firewall Enforcement	Application Control	Identity Aware
Excellent	100%	100%	100%

Using the default policy, the 8250 blocked 99.1% of attacks against client applications and 98.9% overall. Sourcefire 8250 5.1 correctly identified 100% of our evasion attempts without error.

The product successfully passed 10Gbps of inspected traffic, and in a typical network this could be considered an accurate rating given the headroom available. NSS Labs rates throughput based upon an average of the results from tests: "Real World" Protocol Mix (Perimeter), "Real World" Protocol Mix (Core), and 21 KB HTTP Response respectively.

Sourcefire's management interface has changed significantly from the previous version of Defense Center. Where the older version was somewhat sparse and minimalist, the 5.1 interface is populated with tabs, borders, interactive graphs, and a virtual device display. These GUI changes imbue the management system with many of the same features and capabilities seen in competitive vendor management platforms, while at the same time keeping the overall feel to which users of Sourcefire's Defense Center are accustomed. The management interface is snappy and smooth, allowing for quick navigation to the desired information. Tuning and maintenance have changed little aside from aesthetics, and current users of Sourcefire's Defense Center should find their way around the new interface quickly.

Enterprises looking to update their network defenses with a Next-Generation Firewall can consider Sourcefire's entry into the NGFW market as a solid contender.

2.1 Firewall Policy Enforcement

This section verifies that the DUT is capable of enforcing a specified security policy effectively. NSS Labs' NGFW testing is conducted by incrementally building upon a baseline configuration (simple routing with no policy restrictions and no content inspection) to a complex real world multiple zone configuration supporting many addressing modes, policies, applications, and inspection engines.

With each new security policy, test traffic is passed across the DUT to ensure that only specified traffic is allowed and the rest is denied, and that appropriate log entries are recorded.

The DUT must support stateful firewalling either by managing state tables to prevent "traffic leakage" or as a stateful proxy. The ability to manage firewall policy across multiple interfaces/zones is a required. At a minimum, the DUT must provide a "trusted" internal interface, an "untrusted" external/Internet interface, and one or more DMZ interfaces. In addition, a dedicated management interface is preferred.

Test ID	Test Procedure	Result
3.1.1	Baseline Policy	PASS
3.1.2	Simple Policy	PASS
3.1.3	Complex Policy	PASS
3.1.4	Static NAT	PASS
3.1.5	Dynamic / Hide NAT	PASS
3.1.6	SYN Flood Protection	PASS
3.1.7	Address Spoofing Protection	PASS
3.1.8	Session Hijacking Protection	PASS

2.2 Application Control

An NGFW must provide granular control based upon applications access, not just assigning rules based on ports. This capability is needed to re-establish a secure perimeter where unwanted applications are unable to tunnel over HTTP/S. As such, granular application control is a requirement of NGFW since it enables the administrator to define security policies based upon applications rather than ports alone.

Test ID	Test Procedure	Result
3.2.1	Block Unwanted Applications	100%
3.2.2	Block Specific Action	100%

Tests determined that Sourcefire 8250 5.1 correctly enforced complex outbound and inbound policies consisting of several rules, objects, and applications. It was verified that the device successfully determined the correct application and took the appropriate action based upon the policy.

The Sourcefire 8250 is capable of enforcing application control on any port, including non-standard ports for a particular application. This requires a specific rule to be created by the administrator, identifying and labeling the application from one of the many applications/protocols recognized by the device. Rule creation is achieved quickly, with the ability to select applications and protocols from searchable drop-down list, and is as easy as filling in the appropriate blanks and applying the policy.

2.3 User/Group ID Aware Policies

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

Test ID	Test Procedure	Result
3.3.1	Users Defined via NGFW Integration with Active Directory	100%
3.3.2	Users Defined in NGFW DB (where AD integration is not available)	N/A

Integrating the Sourcefire 8250 with the existing NSS Labs test network Active Directory implementation was easy, though it required some configuration changes to the network. While many vendor solutions will allow their FW/NGFW devices to communicate with Active Directory through their protected ports, all AD information is gathered through the dedicated management port of the Sourcefire 8250. For environments with a segregated network, communication traffic will need to be explicitly allowed between the Sourcefire 8250 management network and the network housing the AD environment. Once that was done, the Sourcefire 8250 5.1 was able to correctly enforce complex outbound and inbound policies consisting of many rules, objects and applications, based on AD user/group policies. NSS engineers verified that the device successfully identified the users and groups and took the appropriate action based upon the firewall policy.

2.4 Intrusion Prevention

In order to represent accurately the protection that is likely to be achieved by a typical enterprise, NSS Labs evaluates the device under test (DUT) using the pre-defined default or recommended configuration that ships with the product "out-of-the-box".

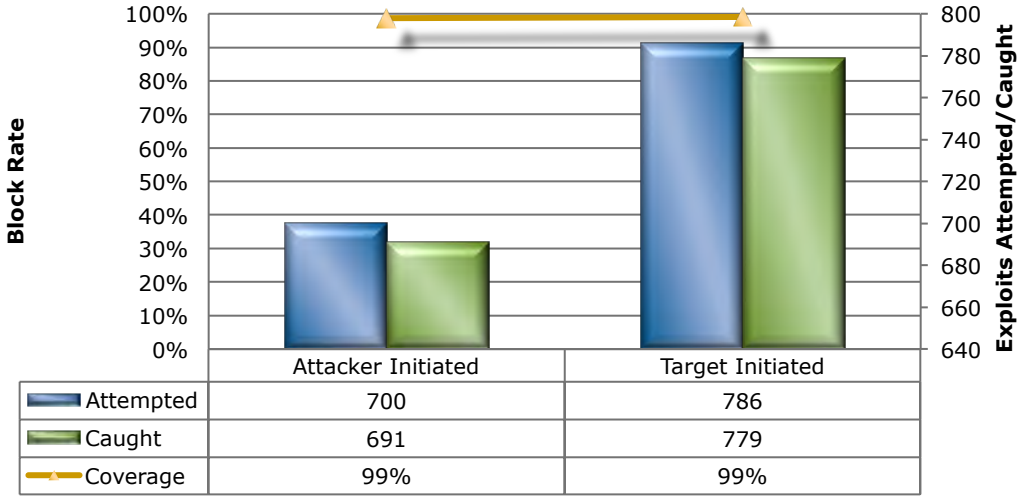
Live Exploit Testing: NSS Labs' security effectiveness testing leverages deep expertise of our engineers utilizing multiple commercial, open source and proprietary tools as appropriate, with over 1,400 live exploits, providing the industry's most comprehensive test. Most notable, all of the live exploits and payloads in our test have been validated in our lab 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 installed
- a system is rendered unresponsive
- etc.

Configuration	Total Number of Exploits Run	Total Number Blocked	Block Percentage
Default Configuration	1,486	1,470	98.9%

2.4.1 Coverage by Attack Vector

Because a failure to block attacks could result in significant compromise and impact to critical business systems, Next-Generation Firewalls should be evaluated against a broad set of exploits. Exploits can be categorized into two groups: *attacker-initiated* and *target initiated*. Attacker-initiated exploits are threats executed remotely against a vulnerable application and/or operating system by an individual while target-initiated exploits are initiated by the vulnerable target. In target-initiated exploits, the attacker has little or no control as to when the threat is executed.



	Attacker Initiated	Target Initiated
Attempted	700	786
Caught	691	779
Coverage	99%	99%

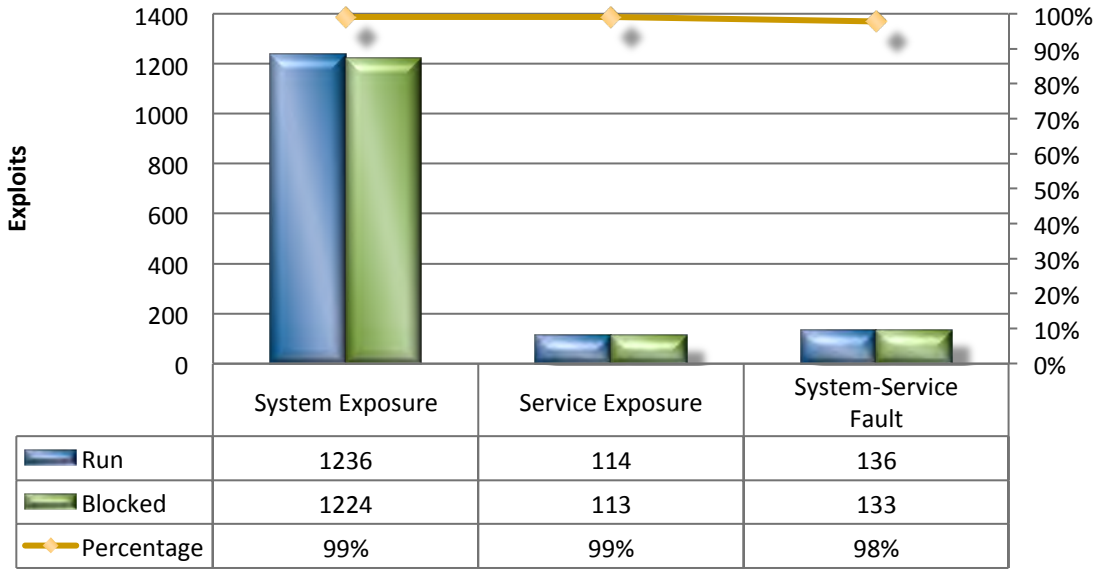
Figure 1: Coverage by Attack Vector

2.4.2 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, 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 (often target initiated) 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	1236	114	136
Blocked	1224	113	133
Percentage	99%	99%	98%

Figure 2: Product Coverage by Impact

2.4.3 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. Further details are available in the NSS Labs *Exposure Report* for this product.

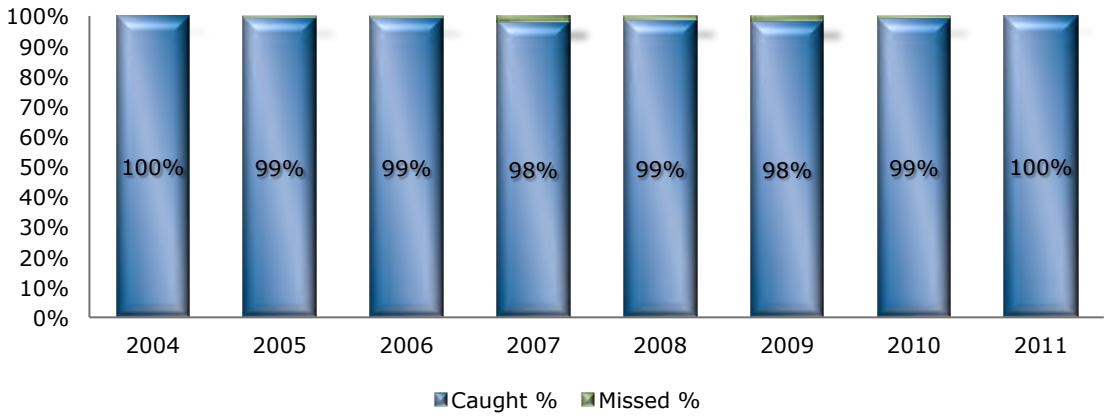


Figure 3: Product Coverage by Date

2.4.4 Coverage by Target Vendor

The NSS Labs 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 Sourcefire 8250 for the top 5 vendor targets

(out of more than 70) represented in this round of testing. Further details are available in the NSS Labs *Exposure Report* for this product.

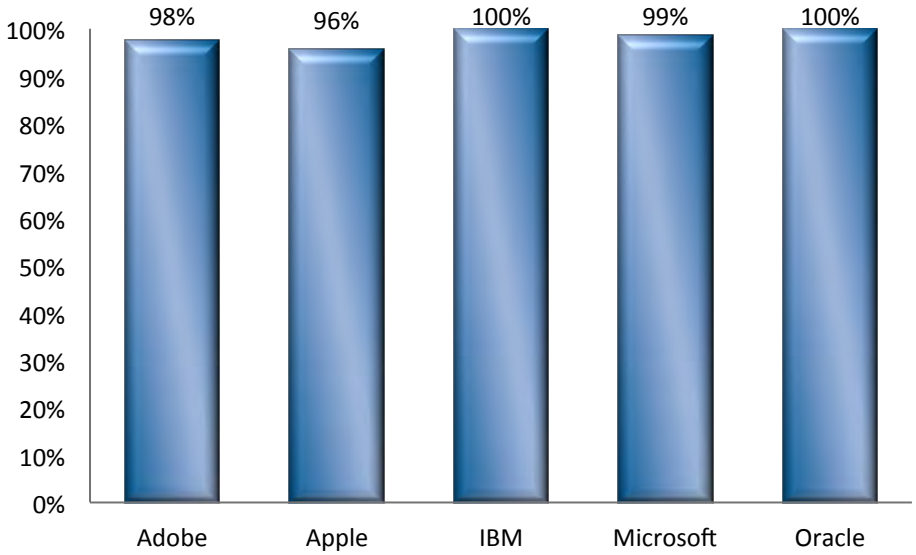


Figure 4: Product Coverage by Target Vendor

2.4.5 Coverage by Result

These tests determine the protection provided against different types of exploits based on the intended action of those exploits, e.g. arbitrary execution, buffer overflow, code injection, cross-site scripting, directory traversal, privilege escalation, etc. Further details are available in the NSS Labs *Exposure Report* for this product.

2.4.6 Coverage by Target Type

These tests determine the protection provided against different types of exploits based on the target environment, e.g. Web server, Web browser, database, ActiveX, Java, browser plugins, etc. Further details are available in the NSS Labs *Exposure Report* for this product.

2.4.7 Attack Leakage

Unlike NIPS, a Firewall must never allow traffic to pass without inspection in "bypass" mode. The Sourcefire 8250 5.1 will drop new connections when resources (such as state table memory) are low, or when traffic loads exceed the device capacity. This will theoretically block legitimate traffic, but maintain state on existing connections (preventing evasion). This is the correct response and prevents attack leakage.

2.5 Resistance to Evasion

Description	IP Packet Fragmentation	TCP Stream Segmentation	RPC Fragmentation	URL Obfuscation	HTML Evasion	FTP Evasion	TOTAL
Sourcefire 8250 5.1	100%	100%	100%	100%	100%	100%	100%

Resistance to known evasion techniques was perfect, with the Sourcefire 8250 5.1 achieving a 100% score across the board in all related tests. *IP fragmentation, TCP stream segmentation, RPC fragmentation, URL obfuscation, HTML Evasion* and *FTP evasion* all failed to trick the product into ignoring valid attacks. Not only were the fragmented and obfuscated attacks blocked successfully, but all of them were also decoded accurately.

3 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.

3.1 Connection Dynamics – Concurrency and Connection Rates

The aim of these tests is to stress the detection engine and determine how the sensor copes with large numbers 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 firewall is causing unacceptable increase in open connections on the server-side.

Excessive response time for HTTP transactions/SMTP sessions - latency within the firewall is causing excessive delays and increased response time to the client.

Unsuccessful HTTP transactions/SMTP sessions – normally, there should be zero unsuccessful transactions. Once these appear, it is an indication that excessive latency within the firewall is causing connections to time out.

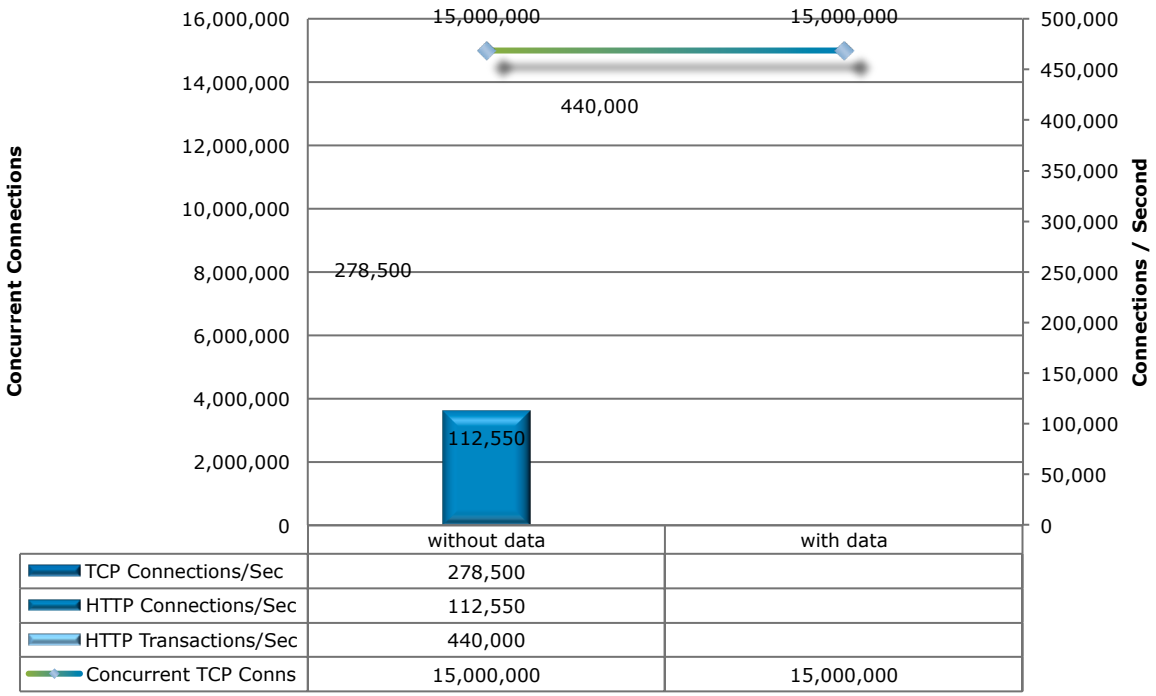


Figure 5: Concurrency and Connection Rates

3.2 HTTP Connections per Second and Capacity

These tests aim to stress the HTTP detection engine in order to determine how the sensor copes with detecting and blocking exploits under network loads of varying average packet size and varying connections per second. By creating genuine session-based traffic with varying session lengths, the sensor is forced to track valid TCP sessions, thus ensuring a higher workload than for simple packet-based background traffic.

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

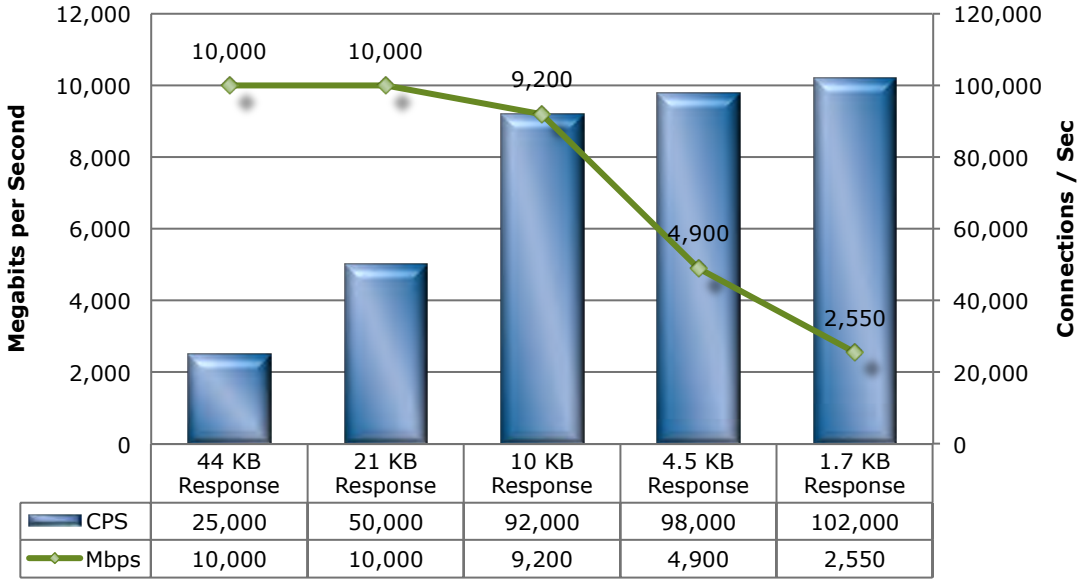


Figure 6: HTTP Connections per Second and Capacity

3.3 Application Average Response Time – HTTP

4.3	Application Average Response Time - HTTP (at 90% Max Load)	Milliseconds
4.3.1	2.500 Connections Per Second – 44Kbyte Response	0.99
4.3.2	5,000 Connections Per Second – 21Kbyte Response	1.01
4.3.3	10,000 Connections Per Second – 10Kbyte Response	1.59
4.3.4	20,000 Connections Per Second – 4.5Kbyte Response	1.51
4.3.5	40,000 Connections Per Second – 1.7Kbyte Response	1.56

3.4 HTTP Connections per Second and Capacity (With Delays)

Typical user behavior introduces delays between requests and reponses, e.g. “think time”, as users read web pages and decide which links to click next. This group of tests is identical to the previous group except that these include a 10 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.

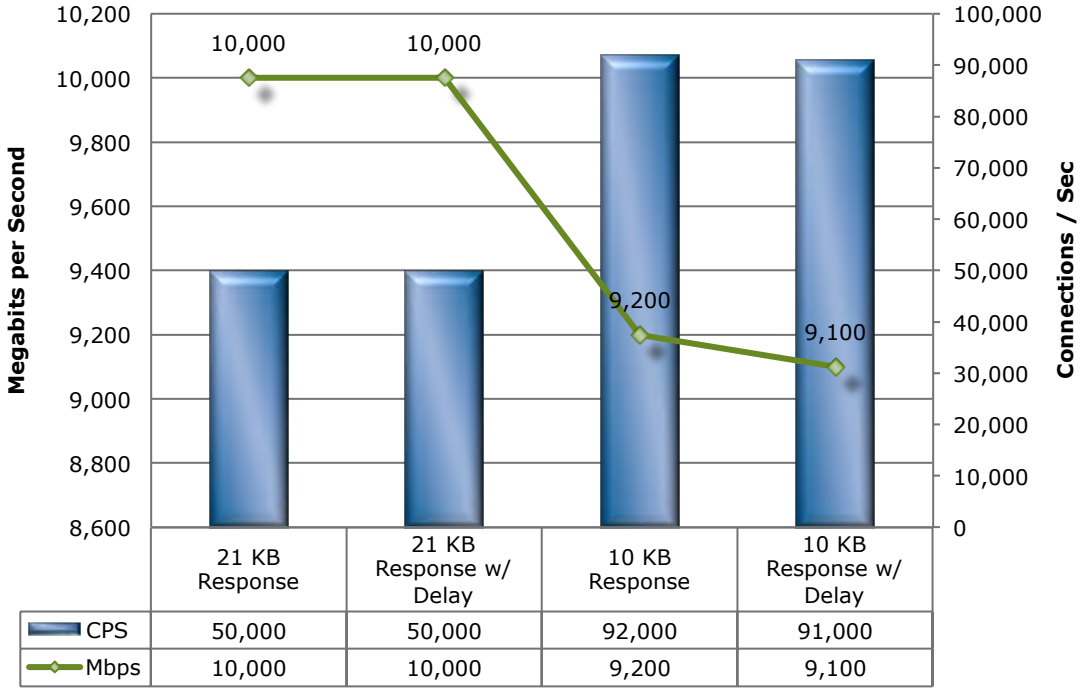


Figure 7: HTTP Connections per Second and Capacity (With Delays)

3.5 UDP Throughput

The aim of this test is to determine the raw packet processing capability of each in-line port pair of the device only.

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 detection engine to do in the way of protocol analysis (although each vendor will be required to write a signature to detect the test packets to ensure that they are being passed through the detection engine and not "fast-tracked" from the inbound to outbound port).

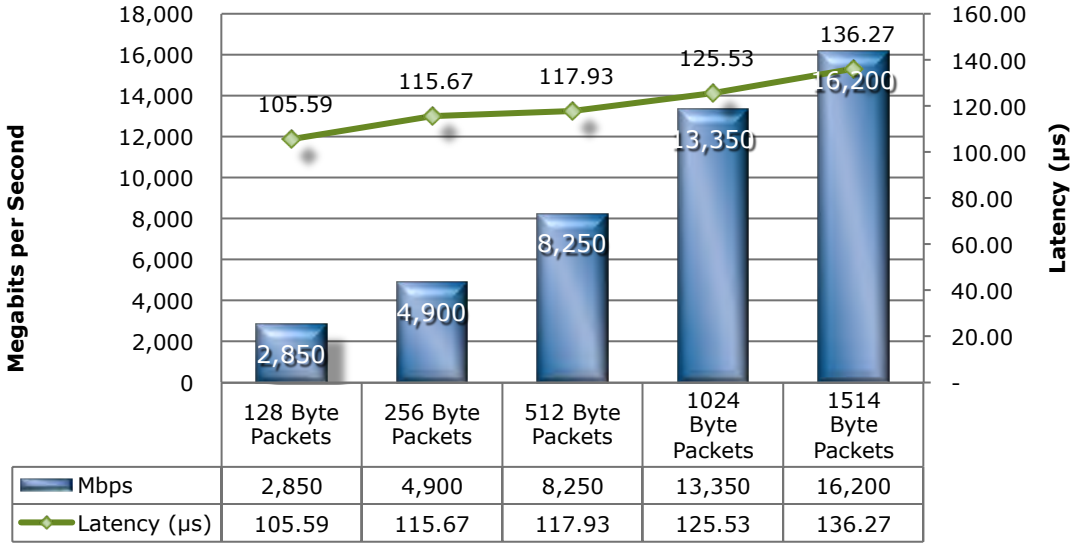


Figure 8: UDP Throughput

3.6 Latency – UDP

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

4.6	Latency - UDP	Microseconds
4.6.1	128 Byte Packets	105.59
4.6.2	256 Byte Packets	115.67
4.6.3	512 Byte Packets	117.93
4.6.4	1024 Byte Packets	125.53
4.6.5	1514 Byte Packets	136.27

3.7 Real-World Traffic Mixes

The aim of this test is to measure 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 location of the device under test to

reflect real use cases. For details about real world traffic protocol types and percentages, see the NSS Labs NGFW Test Methodology, available at www.nsslabs.com.

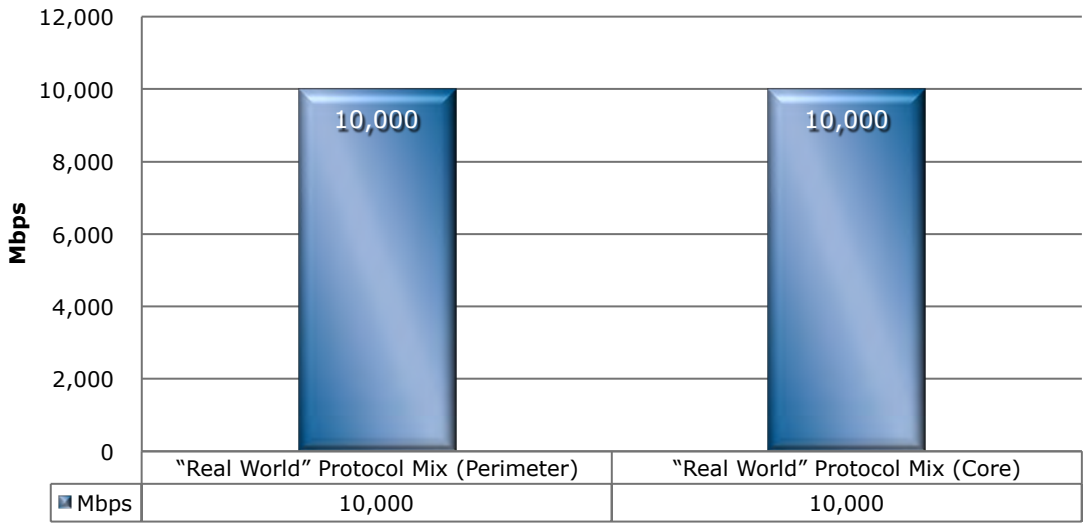


Figure 9: Real-World Traffic Mixes

4 Stability & Reliability

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

The DUT is required to remain operational and stable throughout these tests, and to block 100 per cent of previously blocked traffic, raising an alert for each. If any non-allowed traffic passes successfully - caused by either the volume of traffic or the DUT failing open for any reason - this will result in a FAIL.

Test ID	Test Procedure	Result
5.1	Blocking Under Extended Attack	PASS
5.2	Passing Legitimate Traffic Under Extended Attack	PASS
5.3	Behavior Of The State Engine Under Load	PASS
5.3.1	Attack Detection/Blocking - Normal Load	PASS
5.3.2	State Preservation - Normal Load	PASS

5.3.3	Pass Legitimate Traffic - Normal Load	PASS
5.3.4	State Preservation - Maximum Exceeded	PASS
5.3.5	Drop Traffic - Maximum Exceeded	PASS
5.4	Protocol Fuzzing & Mutation	PASS
5.5	Power Fail	PASS
5.6	Redundancy	YES
5.7	Persistence of Data	PASS

5 Management & Configuration

5.1 General

In addition to the specific tests noted below, NSS has executed an in-depth technical evaluation of all the main features and capabilities of the enterprise management system offered by the vendor. This will typically be offered as an extra-cost option.

Question	Answer
Transparent Mode - Is DUT capable of running in transparent bridge mode, with no IP address assigned to detection ports. Detection ports should ignore all direct connection attempts.	Yes
Routed Mode - Is DUT capable of running in full routed mode, with IP address assigned to detection ports.	Yes
Management Port - Does DUT feature a dedicated management port, separate from detection ports. Although this is the preferred configuration, lack of a management port (requiring DUT to be managed via one of the detection ports) will not be an issue providing management connection and communication is securely encrypted.	Yes
Management Protocol – Is connection from management console to DUT protected by a minimum of a user name/password combination or multi-factor authentication system, and are all communications securely encrypted. Where a three-tier management architecture is employed, all communication between console and management server(s), and between management server(s) and sensor(s) should be securely encrypted.	Yes
Authentication – Is access to management console protected by a granular user authentication system which allows for separation of read only and read-write access, preventing users who require reporting access only from modifying device parameters, etc. No access to administrative functions should be permitted (using either direct or centralized administration capabilities) without proper authentication.	Yes – User creation and pre-defined roles exist in the Defense Center for managing access

Enterprise Authentication – Is access to management console protected by a granular user authentication system that allows for restriction of individual users to specific devices, ports, reports, and security policies. Authenticated users should be unable to access devices/ports/policies/alerts/reports/etc. restricted to other users of the system.	Yes – User creation and pre-defined roles exist in the Defense Center for managing access
Direct Device Management – Is direct access to the DUT provided (either via command line or Web interface) for single-device management.	Yes, through both command line and a web-based GUI.
Centralized Device Management – Is a centralized management system provided to manage one or more sensors from a single point, including centralized device configuration, policy definition, alert handling and reporting for all sensors under the control of the management system. This should be scalable to large numbers of sensors.	Yes – Defense Center system manages multiple sensors, including centralized policy, alerting, and reporting
Pass-Through Mode – Is it possible to place the DUT into a mode whereby all traffic is allowed to pass through the device, but data will be logged according to the policy in place at the time (thus, the DUT will log alerts and state whether the packets would have been dropped, session terminated, etc., but without enforcing those actions on the traffic processed). This should be via a single system-wide operation via the management console or DUT command line (i.e. it is not permitted to achieve this by requiring that all BLOCK signatures be amended to LOG ONLY, or by switching policies - it must be achieved without affecting the current policy in force).	Yes
IPS Signature Update - Can vendor demonstrate access to a vulnerability research capability (either in-house or via a recognized third-party) that is able to provide timely and accurate signature updates at regular intervals.	Yes – Sourcefire Vulnerability Research Team (VRT) in-house research team
Secure Device Registration – Is initial registration of DUT to central management console performed in a fully secure manner (it is permitted to offer a less secure/rapid option, but this should not be the default).	Yes

5.2 Policy

Question	Answer
Device Configuration - Does management system provide the means to configure one or more sensors from a central location, assigning signatures, sensor settings, etc.	Yes – Sensor grouping in Defense Center (DC)
Policy Definition - Does management system provide the means to define and save multiple security policies, consisting of: general sensor configuration, system-wide parameters, firewall policy, signatures enabled/disabled, actions to take when malicious traffic discovered	Yes – DC provides ability for grouping of policies and sensors individually
Recommended Settings - Does vendor provide a default policy or suite of recommended IPS settings which comprises the optimum configuration for a typical network (including which signatures are enabled/disabled, which are enabled in blocking mode, required actions, etc.)	Yes
Custom Attack Signatures – Is it possible for the administrator to define custom IPS signatures for use in standard policies? If so, what for do these take (Snort compatible, etc.)	Yes - Snort

Bulk Operations – Is it possible to search quickly and easily for individual signatures or groups/classes of signatures, and subsequently to apply one or more operations to an entire group in a single operation (for example, to enable or disable a group of signatures, or to switch a group from block mode to log mode, etc.)	Yes – built in search in DC interface
Granularity – Is the DUT capable of blocking or creating exceptions based on IP address, application, user/group ID, protocol, VLAN tag, etc. (i.e. never block HTTP traffic between two specific IP addresses, always block FTP traffic to one specific IP address, etc.).	Yes – Drill down into events and "right-click" blocking of events in near real time
Policy Association - Once policies have been defined, is it possible to associate them with specific devices or groups of devices.	Yes
Inheritance – Is it possible to create groups and sub-groups of devices such that sub-groups can inherit certain aspects of configuration and policy definition from parent groups.	Yes – through grouping in the DC, sensors can inherit configuration and policy from parent lists
Virtualization - Once policies have been defined, is it possible to associate them with specific "virtual" devices or groups of devices, comprising an entire DUT, individual ports, port groups, IP address range, subnet or VLAN.	Yes
Policy Deployment - Once policies have been defined, is it possible to distribute them to the appropriate device(s), virtual device(s), or groups of devices in a single operation.	Yes
Policy Auditing - Are changes to policies logged centrally. Log data should include at a minimum the date/time the changes were made, and the identity of the user who made them. If possible the system should record the actual changes.	Yes
Policy Version Control - Are changes to policies recorded by saving a version of the policy before each change. Is it possible to roll back to a previous version of any policy via a single operation.	Yes

5.3 Alert Handling

Question	Answer
Generic Log Events - Does DUT record log entries for the following events: Detection of malicious traffic, termination of a session, successful authentication by administrator, unsuccessful authentication by administrator, policy changed, policy deployed, hardware failure, power cycle	Yes
Log Location - Are log events logged on the DUT initially, in a secure manner, and subsequently transmitted to a central console/management server for permanent storage.	Yes
Communication Interruption - Where communications between sensor and console/management server are interrupted, how much storage capacity is available on the DUT to store log data (in days/weeks). If it is not possible to restore communication in a timely manner, once the local logs are full, the DUT should either (1) continue passing traffic and overwrite oldest log entries, or (2) stop passing traffic. Which option is employed, and is it configurable by the administrator.	Potentially years of storage available. Overwrite old logs, configurable in the DC

Log Flooding – Are mechanisms in place (aggregation) to prevent the DUT from flooding the management server/console with too many events of the same type in a short interval. Is it possible to disable aggregation/flood protection completely for testing purposes to ensure NSS can see every individual alert.	Yes – default aggregation of like events
Alerts - Does DUT record log entries each time it detects malicious traffic. What information is recorded?	Yes, Full payloads, Configurable
Alert Accuracy - Does DUT record log entries that are accurate and human readable without having to use additional reference material. The DUT should attempt to minimize the number of alerts raised for a single event wherever possible.	Yes
Centralized Alerts – Are all alerts delivered to, and handled by, a single, central, management console. Is it possible to view all alerts globally, or select alerts from individual devices (logical or physical).	Yes – DC handles and allows for review of events
Alert Delivery Mechanism - Does the DUT deliver alerts in a timely manner to a central database for permanent storage, central console for a real-time display, and SMTP server for e-mail alerts.	Yes
Alert Actions - On detecting malicious traffic, what actions can the DUT perform e.g. Ignore, Log only, Allow, Block, Drop packet (no reset), Drop session (no reset), E-mail administrator, send TCP reset (or ICMP redirect) to source only, Send TCP reset (or ICMP redirect) to destination only, Send TCP reset (or ICMP redirect) to both source and destination, Reconfigure firewall, Reconfigure switch to isolate/quarantine offending port, Page administrator	All of these actions are available, with configuration.
Forensic Analysis - Can DUT capture individual packets, a range of packets, or an entire session where required (globally, or on a rule-by-rule basis)	Yes – per rule set basis
Summarize Alerts – Can the central console provide the ability to select a particular piece of data from an alert and summarize on that data field (i.e. select a source IP address and view all alerts for that source IP). Alternatively, it should be possible to construct data filters manually in a search form and summarize on the specified search criteria. The preferred scenario is to offer both of these options.	Yes – Built in search option allows for search on a variety of fields, as well as drill-down into individual events, sortable by several data points
View Alert Detail – Does the central console provide the ability to select an individual alert and view the following information at a minimum: Detailed alert data, Detailed exploit data (description of the exploit research), Signature/rule, Remediation data/preventative action	Yes - Detailed alert data, Detailed exploit data (description of the exploit research), Signature/rule, Remediation data/preventative action
View Policy - Having selected an alert, does the system provide the ability to access directly the policy and rule that triggered the event in order to view and/or modify the policy for further fine-tuning.	Yes
View Packet Contents – Does the central console provide the ability to select an individual alert and view the contents of the trigger packet or context data for the exploit.	Yes
Alert Suppression - The central console should provide the ability to create exception filters based on alert data to eliminate further alerts which match the specified criteria (i.e. same alert ID from same source IP). This does not disable detection, logging or blocking, but merely excludes alerts from the console display.	Yes – correlation on like events, same alert ID from same source IP, etc.
Correlation (Automatic) – Does the system provide the means to infer connections between multiple alerts and group them together as incidents automatically.	Yes
Correlation (Manual) – Does the system provide the means for the administrator to infer connections between multiple alerts and group them together as incidents manually.	Yes

Incident Workflow – Does the system provide the ability to annotate and track incidents to resolution.	Yes

5.4 Reporting

Question	Answer
Centralized Reports – Is the system capable of reporting on all alerts from a single, central, management console. From that console, is it possible to report all alerts globally, or to report on alerts from individual devices (logical or physical).	Yes
Built In Reports - Does system provide built in reports covering typical requirements such as list of top attacks, top source/destination IP addresses, top targets, etc.	Yes – both exportable and as widgets on the DC
Custom Reports – Does the system offer a report generator providing the ability to construct complex data filters in a search form and summarize alerts on the specified search criteria.	Yes
Saved Reports - Having defined a custom report filter, is it possible to save it for subsequent recall.	Yes
Scheduled Reports – Is it possible to schedule saved reports for regular unattended runs. If so, how is the output saved (as HTML or PDF, for example). Is it possible to publish reports to a central FTP/Web server, and/or e-mail reports to specified recipients.	Yes, configurable
Log File Maintenance - Does system provide for automatic rotation of log files, archiving, restoring from archive, and reporting from archived logs.	Yes

6 Total Cost of Ownership (TCO)

Next-Generation Firewall solutions can be complex projects 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 signature updates.)
- **Installation** – the time required to unbox and rack the device, configure it, put it into the network, apply updates and patches, initial tuning, and set up desired logging and reporting.
- **Upkeep** – the time required to apply periodic updates and patches from vendors, including hardware, software, and protection (signature/filter/rules) updates.
- **Tuning** – the time required to configure the policy such that the best possible protection is applied while reducing or eliminating false alarms and false positives. NSS Labs assumes enterprises will use pre-defined vendor policies and therefore eliminating tuning.

6.1 Labor per Product (in Hours)

This table estimates the annual labor required to maintain each device. NSS Labs' assumptions are based upon the time required by an experienced security engineer ($75 per hour fully loaded,) allowing NSS to keep the hourly wage cost constant, and measure only the difference in time required to tune. Readers should substitute their own costs to obtain accurate TCO figures.

Product	Installation (Hrs)	Upkeep / Year (Hrs)	Tuning / Year (Hrs)
Sourcefire 8250 5.1	8	24	24

6.2 Purchase Price and Total Cost of Ownership

Calculations are based on vendor-provided list price. 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 enterprise management solutions will be extra.

Product	Purchase	Maintenance / year	1 Year TCO	2 Year TCO	3 Year TCO
Sourcefire 8250 5.1	$219,490	$31,762	$255,452	$290,813	$326,175

- Year One TCO was determined by multiplying the Labor Rate ($75 per hour fully loaded) x (Installation + Upkeep + Tuning) and then adding the Purchase Price + Maintenance.
- Year Two TCO was determined by multiplying the Labor Rate ($75 per hour fully loaded) x (Upkeep + Tuning) and then adding Year One TCO.
- Year Three TCO was determined by multiplying the Labor Rate ($75per hour fully loaded x (Upkeep + Tuning) and then adding Year Two TCO.

6.3 Value: Cost per Mbps and Exploit Blocked

There is a clear difference between price and value. The least expensive product does not necessarily offer the greatest value if it blocks fewer exploits than its competitors. The best value is a product with a low TCO and high level of secure throughput (security effectiveness x performance).

The following table illustrates the relative cost per unit of work performed: Mbps-Protected

Product	Protection	Throughput	3 Year TCO	Price / Mbps-Protected
Sourcefire 8250 5.1	98.9%	10,000	$326,175	$33

Price per Protected Mbps was calculated by taking the Three-Year TCO and dividing it by the product of Protection x Throughput. Three-Year TCO/(Protection x Throughput) = Price/Mbps-Protected.

7 Detailed Product Scorecard

The following chart depicts the status of each test with quantitative results where applicable. A separate product Exposure Report details specific vulnerabilities that are not protected.

Test ID	Description	Result
3	Security Effectiveness	
3.1	Firewall Policy Enforcement	
3.1.1	Baseline Policy	PASS
3.1.2	Simple Policy	PASS
3.1.3	Complex Policy	PASS
3.1.4	Static NAT	PASS
3.1.5	Dynamic / Hide NAT	PASS
3.1.6	SYN Flood Protection	PASS
3.1.7	Address Spoofing Protection	PASS
3.1.8	Session Hijacking Protection	PASS
3.2	Application Control	
3.2.1	Block Unwanted Applications	PASS
3.2.2	Block Specific Action	PASS
3.3	User / Group ID Aware Policies	
3.3.1	Users Defined via NGFW Integration with Active Directory	PASS
3.3.2	Users Defined in NGFW DB (Alternate to 3.3.1)	N/A
3.4	Intrusion Prevention	
3.4.1	Coverage By Attack Vectors	
3.4.1.1	Attacker Initiated	99%
3.4.1.2	Target Initiated	99.1%
3.4.1.3	Combined Total	98.9%
3.4.2	Coverage By Impact Type	
3.4.2.1	System Exposure	99%
3.4.2.2	Service Exposure	99%
3.4.2.3	System or Service Fault	98%
3.4.3	Coverage by Date	Contact NSS
3.4.4	Coverage by Target Vendor	Contact NSS
3.4.5	Coverage by Result	Contact NSS
3.4.6	Coverage by Target Type	Contact NSS
3.4.7	Attack Leakage	PASS
3.5	Evasion	100%
3.5.1	Packet Fragmentation	100%

3.5.1.1	Ordered 8 byte fragments	100%
3.5.1.2	Ordered 24 byte fragments	100%
3.5.1.3	Out of order 8 byte fragments	100%
3.5.1.4	Ordered 8 byte fragments, duplicate last packet	100%
3.5.1.5	Out of order 8 byte fragments, duplicate last packet	100%
3.5.1.6	Ordered 8 byte fragments, reorder fragments in reverse	100%
3.5.1.7	Ordered 16 byte frags, fragment overlap (favor new)	100%
3.5.1.8	Ordered 16 byte frags, fragment overlap (favor old)	100%
3.5.1.9	Out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery	100%
3.5.2	Stream Segmentation	100%
3.5.2.1	Ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums	100%
3.5.2.2	Ordered 1 byte segments, interleaved duplicate segments with null TCP control flags	100%
3.5.2.3	Ordered 1 byte segs, interleaved duplicate segments with requests to resync sequence numbers mid-stream	100%
3.5.2.4	Ordered 1 byte segments, duplicate last packet	100%
3.5.2.5	Ordered 2 byte segments, segment overlap (favor new)	100%
3.5.2.6	Ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers	100%
3.5.2.7	Out of order 1 byte segments	100%
3.5.2.8	Out of order 1 byte segments, interleaved duplicate segments with faked retransmits	100%
3.5.2.9	Ordered 1 byte segments, segment overlap (favor new)	100%
3.5.2.10	Out of order 1 byte segs, PAWS elimination (interleaved dup segs with older TCP timestamp options)	100%
3.5.2.11	Ordered 16 byte segs, seg overlap (favor new (Unix))	100%
3.5.3	RPC Fragmentation	100%
3.5.3.1	One-byte fragmentation (ONC)	100%
3.5.3.2	Two-byte fragmentation (ONC)	100%
3.5.3.3	All fragments, including Last Fragment (LF) will be sent in one TCP segment (ONC)	100%
3.5.3.4	All frags except Last Fragment (LF) will be sent in one TCP segment. LF will be sent in separate TCP seg (ONC)	100%
3.5.3.5	One RPC fragment will be sent per TCP segment (ONC)	100%
3.5.3.6	One LF split over more than one TCP segment. In this case no RPC fragmentation is performed (ONC)	100%
3.5.3.7	Canvas Reference Implementation Level 1 (MS)	100%
3.5.3.8	Canvas Reference Implementation Level 2 (MS)	100%
3.5.3.9	Canvas Reference Implementation Level 3 (MS)	100%
3.5.3.10	Canvas Reference Implementation Level 4 (MS)	100%
3.5.3.11	Canvas Reference Implementation Level 5 (MS)	100%

3.5.3.12	Canvas Reference Implementation Level 6 (MS)	100%
3.5.3.13	Canvas Reference Implementation Level 7 (MS)	100%
3.5.3.14	Canvas Reference Implementation Level 8 (MS)	100%
3.5.3.15	Canvas Reference Implementation Level 9 (MS)	100%
3.5.3.16	Canvas Reference Implementation Level 10 (MS)	100%
3.5.4	URL Obfuscation	100%
3.5.4.1	URL encoding - Level 1 (minimal)	100%
3.5.4.2	URL encoding - Level 2	100%
3.5.4.3	URL encoding - Level 3	100%
3.5.4.4	URL encoding - Level 4	100%
3.5.4.5	URL encoding - Level 5	100%
3.5.4.6	URL encoding - Level 6	100%
3.5.4.7	URL encoding - Level 7	100%
3.5.4.8	URL encoding - Level 8 (extreme)	100%
3.5.4.9	Premature URL ending	100%
3.5.4.10	Long URL	100%
3.5.4.11	Fake parameter	100%
3.5.4.12	TAB separation	100%
3.5.4.13	Case sensitivity	100%
3.5.4.14	Windows \ delimiter	100%
3.5.4.15	Session splicing	100%
3.5.5	HTML Obfuscation	100%
3.5.5.1	UTF-16 character set encoding (big-endian)	100%
3.5.5.2	UTF-16 character set encoding (little-endian)	100%
3.5.5.3	UTF-32 character set encoding (big-endian)	100%
3.5.5.4	UTF-32 character set encoding (little-endian)	100%
3.5.5.5	UTF-7 character set encoding	100%
3.5.5.6	Chunked encoding (random chunk size)	100%
3.5.5.7	Chunked encoding (fixed chunk size)	100%
3.5.5.8	Chunked encoding (chaffing)	100%
3.5.5.9	Compression (Deflate)	100%
3.5.5.10	Compression (Gzip)	100%
3.5.5.11	Base-64 Encoding	100%
3.5.5.12	Base-64 Encoding (shifting 1 bit)	100%
3.5.5.13	Base-64 Encoding (shifting 2 bits)	100%
3.5.5.14	Base-64 Encoding (chaffing)	100%
3.5.5.15	Combination UTF-7 + Gzip	100%
3.5.6	FTP Evasion	100%

3.5.6.1	Inserting spaces in FTP command lines	100%
3.5.6.2	Inserting non-text Telnet opcodes - Level 1 (minimal)	100%
3.5.6.3	Inserting non-text Telnet opcodes - Level 2	100%
3.5.6.4	Inserting non-text Telnet opcodes - Level 3	100%
3.5.6.5	Inserting non-text Telnet opcodes - Level 4	100%
3.5.6.6	Inserting non-text Telnet opcodes - Level 5	100%
3.5.6.7	Inserting non-text Telnet opcodes - Level 6	100%
3.5.6.8	Inserting non-text Telnet opcodes - Level 7	100%
3.5.6.9	Inserting non-text Telnet opcodes - Level 8 (extreme)	100%
4	Performance	
4.1	Connection Dynamics	
4.1.1	Theoretical Max. Concurrent TCP Connections	15,000,000
4.1.2	Theoretical Max. Concurrent TCP Connections w/Data	15,000,000
4.1.3	Maximum TCP Connections Per Second	278,500
4.1.4	Maximum HTTP Connections Per Second	112,550
4.1.5	Maximum HTTP Transactions Per Second	440,000
4.2	HTTP CPS & Capacity With No Transaction Delays	
4.2.1	2.500 Connections Per Second – 44Kbyte Response	25,000
4.2.2	5,000 Connections Per Second – 21Kbyte Response	50,000
4.2.3	10,000 Connections Per Second – 10Kbyte Response	92,000
4.2.4	20,000 Connections Per Second – 4.5Kbyte Response	98,000
4.2.5	40,000 Connections Per Second – 1.7Kbyte Response	102,000
4.3	Application Average Response Time - HTTP (at 90% Max Load)	Milliseconds
4.3.1	2.500 Connections Per Second – 44Kbyte Response	0.99
4.3.2	5,000 Connections Per Second – 21Kbyte Response	1.01
4.3.3	10,000 Connections Per Second – 10Kbyte Response	1.59
4.3.4	20,000 Connections Per Second – 4.5Kbyte Response	1.51
4.3.5	40,000 Connections Per Second – 1.7Kbyte Response	1.56
4.4	HTTP CPS & Capacity With Transaction Delays	
4.4.1	5,000 Connections Per Second – 21Kbyte Response	50,000
4.4.2	10,000 Connections Per Second – 10Kbyte Response	91,000
4.5	UDP Throughput	Mbps
4.5.1	128 Byte Packets	2,850
4.5.2	256 Byte Packets	4,900
4.5.3	512 Byte Packets	8,250
4.5.4	1024 Byte Packets	13,350
4.5.5	1514 Byte Packets	16,200
4.6	Latency - UDP	Microseconds

4.6.1	128 Byte Packets	106
4.6.2	256 Byte Packets	116
4.6.3	512 Byte Packets	118
4.6.4	1024 Byte Packets	126
4.6.5	1514 Byte Packets	136
4.7	"Real World" Traffic	Mbps
4.7.1	"Real World" Protocol Mix (Perimeter)	10,000
4.7.2	"Real World" Protocol Mix (Core)	10,000
5	Stability & Reliability	
5.1	Blocking Under Extended Attack	PASS
5.2	Passing Legitimate Traffic Under Extended Attack	PASS
5.3	Behavior Of The State Engine Under Load	PASS
5.3.1	Attack Detection/Blocking - Normal Load	PASS
5.3.2	State Preservation - Normal Load	PASS
5.3.3	Pass Legitimate Traffic - Normal Load	PASS
5.3.4	State Preservation - Maximum Exceeded	PASS
5.3.5	Drop Traffic - Maximum Exceeded	PASS
5.4	Protocol Fuzzing & Mutation	PASS
5.5	Power Fail	PASS
5.6	Redundancy	YES
5.7	Persistence of Data	PASS
6	Management & Configuration	
6.1	General	
6.1.1	Transparent Mode	Yes
6.1.2	Routed Mode	Yes
6.1.3	Management Port	Yes
6.1.4	Management Protocol	Yes
6.1.5	Authentication	Yes
6.1.6	Enterprise Authentication	Yes
6.1.7	Direct Device Management	Yes
6.1.8	Centralized Device Management	Yes
6.1.9	Pass-Through Mode	Yes
6.1.10	IPS Signature Update	Yes
6.1.11	Secure Device Registration	Yes
6.2	Policy	
6.2.1	Device Configuration	Yes
6.2.2	Policy Definition	Yes
6.2.3	Recommended Settings	Yes

6.2.4	Custom Attack Signatures	Yes
6.2.5	Bulk Operations	Yes
6.2.6	Granularity	Yes
6.2.7	Policy Association	Yes
6.2.8	Inheritance	Yes
6.2.9	Virtualization	Yes
6.2.10	Policy Deployment	Yes
6.2.11	Policy Auditing	Yes
6.2.12	Policy Version Control	Yes
6.3	Alert Handling	
6.3.1	Generic Log Events	Yes
6.3.2	Log Location	Yes
6.3.3	Communication Interruption	See Section 6.3
6.3.4	Log Flooding	Yes
6.3.5	Alerts	Yes
6.3.6	Alert Accuracy	Yes
6.3.7	Centralized Alerts	Yes
6.3.8	Alert Delivery Mechanism	Yes
6.3.9	Alert Actions	See Section 6.3
6.3.10	Forensic Analysis	Yes
6.3.11	Summarize Alerts	Yes
6.3.12	View Alert Detail	Yes
6.3.13	View Policy	Yes
6.3.14	View Packet Contents	Yes
6.3.15	Alert Suppression	Yes
6.3.16	Correlation (Automatic)	Yes
6.3.17	Correlation (Manual)	Yes
6.3.18	Incident Workflow	Yes
6.4	Reporting	
6.4.1	Centralized Reports	Yes
6.4.2	Built In Reports	Yes
6.4.3	Custom Reports	Yes
6.4.4	Saved Reports	Yes
6.4.5	Scheduled Reports	Yes
6.4.6	Log File Maintenance	Yes
7	Total Cost of Ownership	
7.1	Ease of Use	
7.1.1	Initial Setup (Hours)	8

7.1.2	Time Required for Upkeep (Hours per Year)	24
7.1.3	Time Required to Tune (Hours per Year)	24
7.2	Expected Costs	
7.2.1	Initial Purchase (hardware as tested)	$219,490
7.2.2	Initial Purchase (enterprise management system)	$14,995
7.2.3	Annual Cost of Maintenance & Support (hardware/software)	$31,761.68
7.2.4	Annual Cost of Maintenance & Support (enterprise management system)	$0
7.2.5	Annual Cost of Updates (IPS/AV/etc.)	$0
7.2.6	Installation Labor Cost (@$75/hr)	$600
7.2.7	Management Labor Cost (per Year @$75/hr)	$1,800
7.2.8	Tuning Labor Cost (per Year @$75/hr)	$1,800
7.3	Total Cost of Ownership	
7.3.1	Year 1	$255,452
7.3.2	Year 2	$35,362
7.3.3	Year 3	$35,362
7.3.4	3 Year Total Cost of Ownership	$326,175

Contact Information

NSS Labs, Inc.
6207 Bee Caves Road, Suite 350
Austin, TX 78746 USA
+1 (512) 961-5300
info@nsslabs.com
www.nsslabs.com