



MCAFFEE M-8000 NETWORK IPS
PRODUCT CERTIFICATION



NETWORK INTRUSION PROTECTION SYSTEM (NIPS)
METHODOLOGY VERSION: 5.22
PERFORMANCE: 10-GBPS
JANUARY 16, 2009

Published by NSS Labs.

© 2008 NSS Labs

CONTACT:

5115 Avenida Encinas
Suite H
Carlsbad, CA 92008

Tel: +1.760.412.4627
E-mail: info@nsslabs.com
Internet: <http://www.nsslabs.com>

All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the authors. This report shall be treated at all times as a confidential and proprietary report for internal use only.

Please note that access to or use of this Report is conditioned on the following:

1. The information in this Report is subject to change by NSS Labs without notice.
2. The information in this Report is believed by NSS Labs to be accurate and reliable, but is not guaranteed. All use of and reliance on this Report are at your sole risk. NSS Labs is not liable or responsible for any damages, losses or expenses arising from any error or omission in this Report.
3. NO WARRANTIES, EXPRESS OR IMPLIED ARE GIVEN BY THE NSS LABS. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE DISCLAIMED AND EXCLUDED BY NSS LABS. IN NO EVENT SHALL NSS LABS BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL 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 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 companies mentioned in this report. For PCI-related reports, this does not constitute an endorsement by the PCI Security Standards Council.
6. All trademarks, service marks, and trade names used in this Report are the trademarks, service marks, and trade names of their respective owners, and no endorsement of, sponsorship of, affiliation with, or involvement in, any of the testing, this Report or NSS Labs is implied, nor should it be inferred.

EXECUTIVE SUMMARY

During Q4, 2008 NSS Labs performed comprehensive testing of the McAfee M-8000 (software v4.1) against our Network Intrusion Prevention Systems (NIPS) testing methodology v5.22. This report is based upon empirically validated evidence gathered by NSS Labs during testing.

The M-8000 is a true 10Gbps IPS, providing excellent performance coupled with extremely low latency under all normal traffic conditions. NSS Labs observed throughput in excess of 10Gbps in all UDP tests as well as in both Datacenter and Perimeter “Real World” traffic mixes. The device ably supported over 11Gbps of traffic with the larger HTTP response sizes (21KB) and lower connections per second (5,000 CPS per Gigabit of traffic) found on typical corporate networks.

In our most stressful connections per second tests, the M-8000 was able to achieve 314,000 TCP CPS and exceed 145,000 HTTP CPS, thus enabling us to reach 10Gbps+ maximum throughput with our real-world tests. We also found the M-8000 to be very stable and reliable, coping with our extensive reliability tests with ease.

The security effectiveness of the M-8000 was nearly perfect in all categories, catching 618 of 622 exploits (99.4%). The product had no trouble identifying attacks, even when obfuscated using a wide range of evasion techniques. No false positive alerts were generated by any of our performance test traffic, which comprised a wide variety of protocols including SMB, NetBIOS, FTP, Instant Messaging, and other protocols listed in our real world performance test. In addition, even under our most strenuous tests, resistance to false positives appeared to be high. Thus, the M-8000 requires very little tuning, and is ideally suited to be deployed in high-speed datacenters.

The Network Security Manager UI will be familiar to those already using a McAfee IPS. It is well designed and suited to manage and configure large numbers of NIPS sensors across the enterprise. Alert handling is powerful and flexible. Virtualization and segmentation of policies across various interfaces is possible, enabling granular control of policy. Our one criticism is that the user interface can be overly complicated for network operators, who can quickly become overwhelmed by the extensive tuning and alerting capabilities.

In summary, the M-8000 is a very fast, flexible, and accurate Network Intrusion Prevention System and should be on everyone’s short list for 10 Gbps Internal Datacenter, E-commerce Datacenter, and Corporate Perimeter protection. NSS Labs is pleased to award **NSS Approved** to the McAfee M-8000 Network IPS.

Product:	McAfee M-8000
Version:	Software V4.1
Test Date:	December 2008
Methodology:	Network IPS v5.22
Performance:	10 Gbps
Effectiveness:	99.4%



For the full report and quarterly updates see: <http://www.nsslabs.com/IPS/McAfee-M8000.html>

CONTENTS

1	<i>Introduction</i>	1
2	<i>Test Results</i>	2
2.1	Security Effectiveness	2
2.2	Performance	4
2.3	Stability & Reliability	6
2.4	Management & Usability	7
3	<i>Test Results Scorecard</i>	14
4	<i>The Product Under Test – McAfee M-8000</i>	21
4.1	M-8000 Dual Sensor IPS	21
4.2	M-8000 Hardware	21
4.3	McAfee Network Security Manager	23
5	<i>Test Methodology – Security Effectiveness</i>	24
5.1	Detection Engine	24
5.2	Threat Vector	25
5.3	Evasion	25
5.4	Packet Fragmentation	25
5.5	Stream Segmentation	26
5.6	RPC Fragmentation	27
5.7	URL Obfuscation	27
5.8	FTP Evasion	28
6	<i>Test Methodology – NIPS Performance</i>	29
6.1	Raw Packet Processing Performance (UDP Traffic)	29
6.2	Maximum Capacity	30
6.3	Behavior Of The State Engine Under Load	32
6.4	HTTP Capacity With No Transaction Delays	33
6.5	HTTP Capacity With Transaction Delays	34
6.6	“Real World” Traffic	34
6.7	Latency	36
6.8	User Response Times	37
7	<i>Test Methodology – Stability & Reliability</i>	38
8	<i>Test Methodology – Management & Configuration</i>	40
8.1	Management Port	40
8.2	Management & Configuration - General	40
8.3	Management & Configuration – Policy	42

8.4	Management & Configuration - Alert Handling	43
8.5	Management & Configuration – Reporting	46
	<i>Appendix A: Network IPS Test Environment</i>	<i>48</i>
	<i>Appendix B: Test Infrastructure</i>	<i>49</i>

1 INTRODUCTION

Several trends are converging to make 10 Gbps security a prescient issue for enterprises. Hosted web applications are replacing legacy client-server applications, while at the same time attacks on web apps and databases and even client browsers are on the rise. Indeed, the perimeter continues to become more porous with an increasing number of remote and mobile clients, intranets, partner extranets, and VPNs. And the introduction of cost-effective 10 Gbps networking has risen to enable even greater bandwidth for data – and attacks.

The network core is more vulnerable than ever before. However, protecting the network core is very different from protecting the network perimeter. It requires IPS devices that are not only extremely fast, but also extremely accurate. They must catch attacks traversing 10 Gbps links while producing nearly zero false positives. And they must not degrade network performance or they will never be installed.

Protecting the core demands a new generation of 10 Gbps IPS devices that are an order of magnitude more accurate and manageable than the previous generations of 1 Gbps network IPS.

ABOUT THIS TEST

In Q4 of 2008, NSS Labs began performing the world's first comprehensive 10Gbps Network IPS Group Test. Products were submitted from multiple vendors, including McAfee, who submitted their M-8000 Network Intrusion Prevention appliance.

This report summarizes the results from over 1,500 individual tests and over 5GB of test results collected in our real-world test lab while testing the M-8000. NSS Labs test reports are designed to address the challenges faced by IT professionals in selecting security products. This NSS Labs report provides readers with empirically validated evidence about a product's features and capabilities. Testing and analysis covers several aspects of the security product including:

- ✓ Security Effectiveness
- ✓ Performance
- ✓ Stability and Reliability
- ✓ Management and Usability

As part of its extensive NIPS test methodology, NSS Labs subjects each product to a brutal battery of tests that verify the stability and performance of each device tested, determine the accuracy of its security coverage, and ensure that the device will not block legitimate traffic. If a particular NIPS has been designated as NSS Approved customers can be confident that the device will not significantly impact network performance.

To assess the complex matrix of NIPS performance and security requirements, NSS Labs has developed a specialized lab environment that is able to exercise every facet of a NIPS product. The test suite contains a large variety of individual tests that evaluate the performance, reliability, security effectiveness and usability of NIPS products, providing the most thorough and complete evaluation of NIPS products available anywhere today.

NSS Labs NIPS test methodologies have become the de facto standard for testing in-line NIPS devices. The NSS Approved logo is often an essential item on the list of requirements when purchasing these products.

2 TEST RESULTS

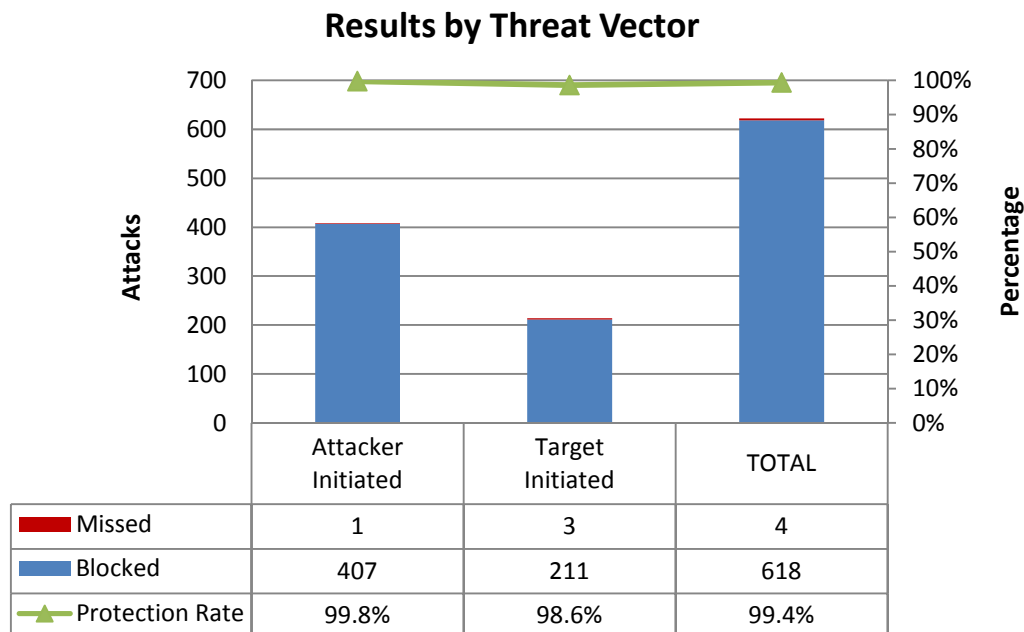
The 10Gbps system consisting of two hardware devices which acted as a single logical unit was installed in our real world test network in-line, with the latest updates, and the default protection policy (with blocking enabled).

2.1 SECURITY EFFECTIVENESS

This section verifies that the DUT is capable of detecting and blocking a wide range of common exploits accurately, while remaining resistant to false positives. All tests are performed initially with no background network load. The tests are then repeated under varying levels and mixes of background traffic to ensure that the results do not vary when handling normal network traffic.

2.1.1 RESULTS BY THREAT VECTORS

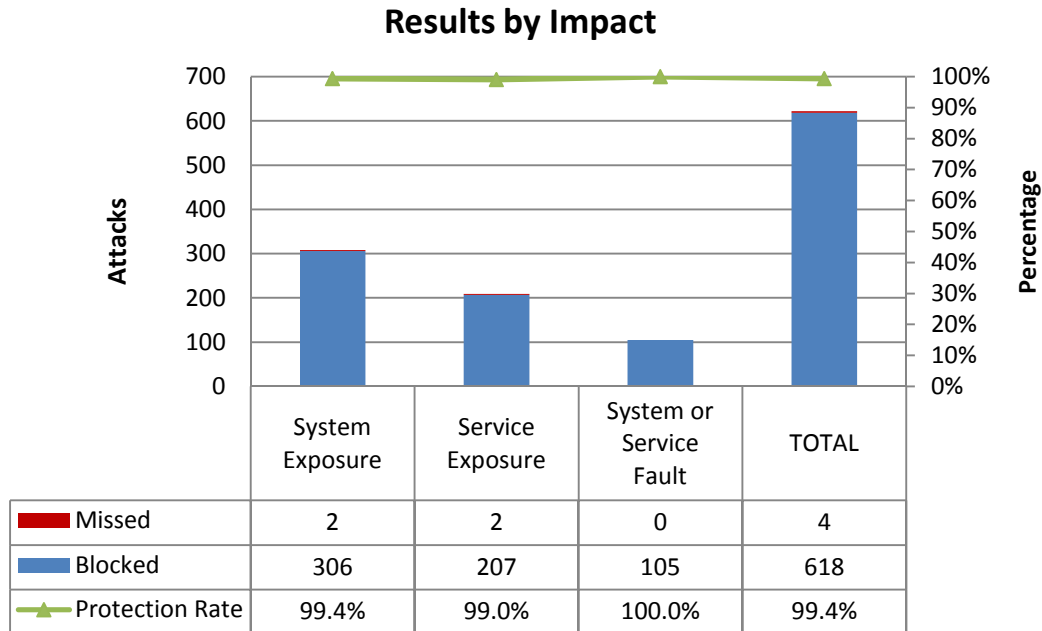
Running a total of 622 exploits, covering a range of operating systems and applications, the M-8000 detected and blocked a total of 618 (99.4%). Of the 408 attacker initiated exploits, the M-8000 missed one, a near-perfect detection rate of 99.8%. Target initiated exploits (server to client) are much more difficult for an IPS to detect, and the M-8000 achieved an impressive score by correctly detecting and blocking 211 out of 214 (98.6%).



2.1.2 RESULTS BY IMPACT

The most serious exploits were those which resulted in a remote system compromise, providing the attacker with the ability to execute arbitrary system level commands. Most exploits in this class that are “weaponized”

will provide the attacker with a fully interactive remote shell on the target client or server. The M-8000 proved near perfect in this highly critical area, detecting 306 of 308 (99.4%).



Slightly less serious are the attacks resulting 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 the 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. Of the 209 exploits in this category, the M-8000 detected 207 (99%).

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 which require administrative action to restart the service or reboot the system. These attacks do not enable the attacker to execute arbitrary commands. However the resulting impact to the business could be severe given that the attacker could crash the protected system or service. Of the exploits in this category, the M-8000 correctly detected and blocked all 105 (100%).

2.1.3 RESISTANCE TO EVASION

Resistance to known evasion techniques was perfect, with the M-8000 achieving a 100% score across the board in all our evasion tests. *IP fragmentation*, *TCP stream segmentation*, *RPC fragmentation*, *URL obfuscation*, and *FTP evasion* all failed to trick the M-8000 into ignoring valid attacks. Not only were the fragmented and obfuscated attacks blocked successfully, but all of them were decoded accurately as well.

2.1.4 ATTACK LEAKAGE

By default, the M-8000 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). There is no right or wrong way to do this. All NIPS

devices have to make the choice whether to risk denying legitimate traffic or allowing malicious traffic once they run low on resources. McAfee has taken the position that security comes first, a hard point to argue against.

Indeed, in our testing, the M-8000 did not leak attacks under any circumstances. Detection and blocking rates at high speed and low speed were identical; with 100 per cent of all attacks that were blocked at low speed being blocked under the most extreme load conditions.

2.2 PERFORMANCE

The M-8000 was tested up to and beyond 10 Gbps, the rated speed of the appliance. Performance varied at the different levels of test depending on the HTTP response sizes and connections per second. The limit of approximately 314,000 TCP connections per second and 145,000 HTTP connections per second provides a balanced weighting of connection rate / payload size.

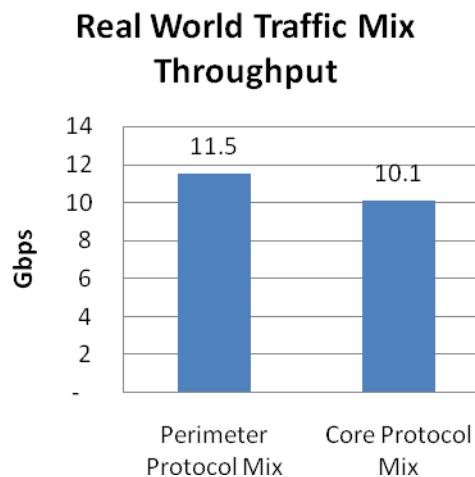
NSS rates IPS performance based upon three metrics:

- Throughput on our real world traffic mix
- HTTP 21KB Response CPS
- 512 byte UDP packet throughput

Based on the data (below), it is NSS Labs' judgment that the M-8000 has earned "true 10-Gigabit" NIPS status in all relevant categories: Network Core Datacenter, E-Commerce Datacenter, and Corporate Perimeter environments.

2.2.1 REAL WORLD TRAFFIC MIX

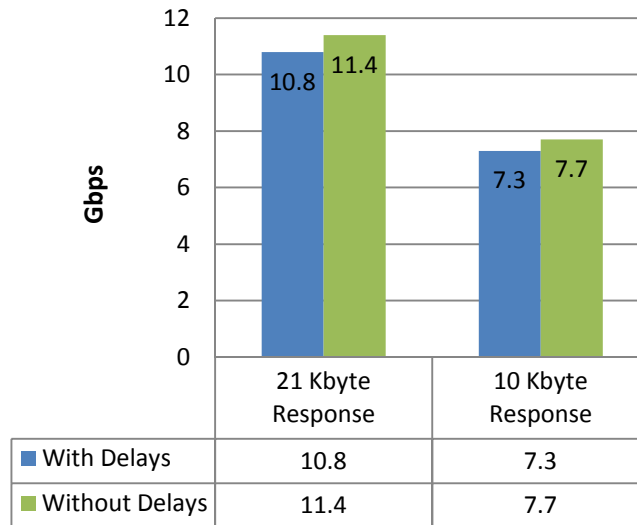
In our Real World Perimeter (11.5 Gbps) & Real World Core (10.1 Gbps) throughput tests, the M-8000 met or exceeded 10 Gbps. For details about real world traffic protocol types and percentages, see section 6.6 below.



2.2.2 HTTP 21KB RESPONSE CPS

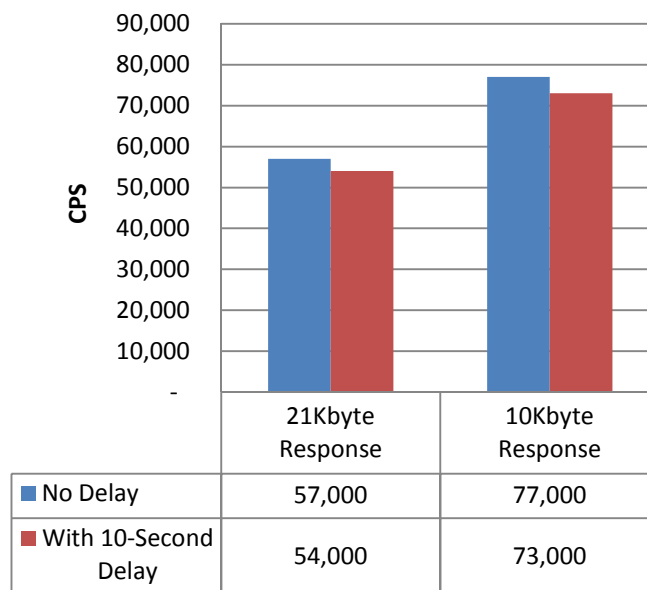
The rated throughput of the M-8000 was 11.4 Gbps with 21 KB response sizes and and 5,000 HTTP connections-per-second per gigabit (50,000 HTTP CPS per 10-Gbps) - which represents a “typical” 10-Gigabit corporate network environment.

Throughput - HTTP Response



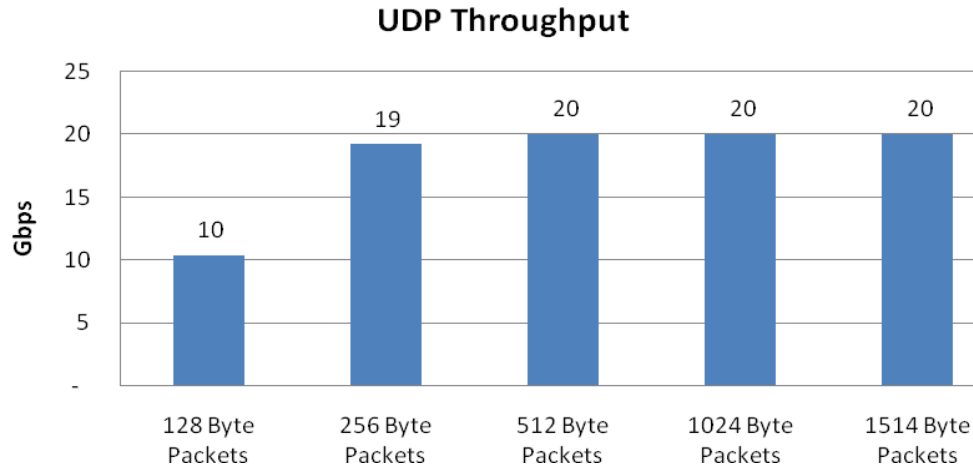
In addition, there was only a minimal drop in CPS when 10-second think time delays were added. This think time simulates real world delays and stresses the product’s state table.

The Impact of Latency & Think Time



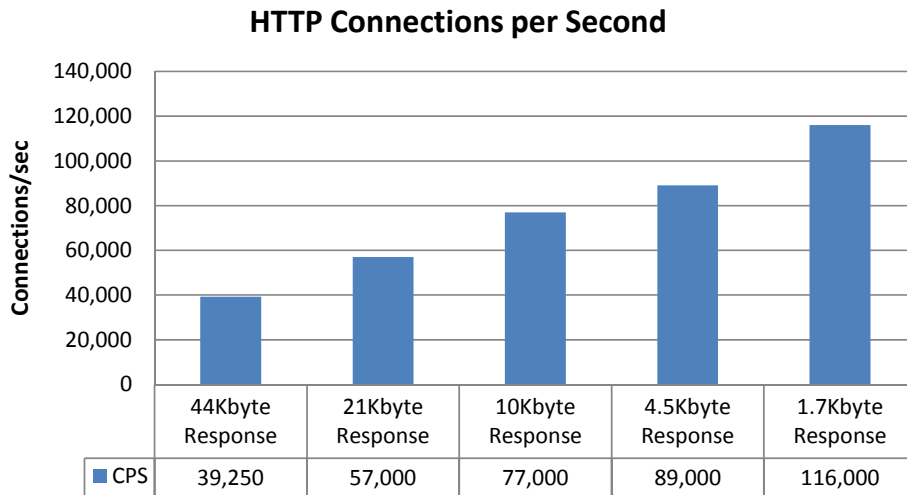
2.2.3 512 BYTE UDP PACKET THROUGHPUT

With 512 byte packets, the M-8000 sustained throughput was rated at 20 Gbps. NSS Labs only validated up to 20 Gbps of UDP traffic since this was a 10Gbps Network IPS test.



2.2.4 HTTP CONNECTIONS & CAPACITY

The following chart depicts the relationship between connection rates, response sizes and throughput.



2.3 STABILITY & RELIABILITY

The M-8000 comes with sixteen (16) small form-factor pluggable 1-Gigabit ports (SFPs) and twelve (12) small form-factor pluggable 10-Gigabit ports (XFPs). SFPs and XFPs are optical transceivers that interface a device’s mother board to either a fiber optic or unshielded twisted pair networking cable. We had some

initial technical difficulty due to a bad batch of SFPs, but once they were replaced, the system worked smoothly. McAfee Certified SFPs and XFPs are definitely a requirement.

Long term stability is particularly important for an in-line NIPS device, where failure can produce network outages. NIPS products that are not able to sustain legitimate traffic (or crash) while under hostile attack will not receive NSS certification.

The M-8000 maintained its composure during our most strenuous tests. It remained operational and stable throughout these tests, and blocked 100 per cent of previously blocked exploits, raising an alert for each. At no time were these exploits successful – due to either the volume of traffic or the NIPS failing open for any reason. In addition, the M-8000 successfully navigated our Protocol Fuzzing and Mutation test suites (based on the ISIC test suite), remaining stable and continuing to block attacks throughout the test

The M-8000 does not leak traffic during reboot, nor when system resources are exhausted. A policy push had a minor impact on the sensor, resulting in temporary increased latency and a slight increase in average HTTP response time. Should power be interrupted, due to a power cycle or other event, all configuration data is retained.

By default, the M-8000 fails closed (does not pass traffic when down). McAfee offers an optional external hardware bypass unit for fail-open operation. The M-8000 failed-open correctly when we tested the fail-open/fail-to-bypass capability, using the hardware bypass unit.

The M-8000 ships with multiple redundant critical components such as fans, power supplies, etc. Due to its unique dual-unit design, it can maintain a high level of performance even when one of the two units fails.

2.4 MANAGEMENT & USABILITY

The Network Security Manager UI will be familiar to those already using a McAfee IPS. 10-Gbps performance and strong resistance to false positives recommends the M-8000 to be deployed in high-speed datacenters, where little tuning is possible, and most operators will be network engineers (and not necessarily security engineers). Unfortunately, the Network Security Manager UI has not been updated to match this new audience and may be overly complex for a non-security focused audience. A simplified UI that focuses on network performance, system health, and major events – with the ability to drill down and create reports as needed – would be better suited. So, while it has the same nice features to be sure, we were disappointed that McAfee did not take this opportunity to refit and update the UI to match the upgrade in product performance and effectiveness.

Selecting one of the pre-defined policies (“All without Audit”), we were able to push the policy out to the desired interfaces. The M-8000 required no subsequent tuning. Alert handling is powerful and flexible, and can be configured to provide automatic correlation when combined with other McAfee products.

The IPS Management Console offers excellent analysis capabilities. The ability to create exceptions and incidents as well as restrict the number of alerts on view was powerful. When coupled with custom reporting, the analysis capabilities will be extremely useful for those needing to adhere with various compliance regimes.

Thus, while we believe the IPS Management Console can benefit from a fresh approach, it continues to be one of the most flexible consoles we have seen in our labs to date, and the M-8000 is probably one of the easiest NIPS products to deploy across a large, distributed network.

2.4.1 CONFIGURATION

The McAfee Network Security Manager has been developed with large-scale distributed deployments in mind. The *administrative domain* is an organizational tool used specifically to group IPS resources so that management of the resources can be delegated to specific IPS users.

An admin domain can contain other admin domains, sensors, sensor interfaces, and sensor sub-interfaces. This administrative domain concept enables enterprises to create a central authority that is responsible for the overall Network Security platform, and to allow this central authority to delegate day-to-day operations of resources to appropriate entities - business units, geographic regions, IT departments, individual security personnel, and so on.

To delegate responsibilities, user accounts are created and allocated a role that defines how the user can interact with the resources in the child admin domain. The Root Admin Domain can be divided into child domains that are large, from a resource perspective, delegating management of all the Network Security Platform resources protecting multiple geographic regions. Or the domains can be very small - a few interfaces on a single sensor, or even a VLAN tag or single CIDR address within a segment of traffic transmitting between two hosts in the protected network.

Child domains can be further broken down into smaller sub-domains in order to provide a very fine degree of management granularity. Administrative domains are graphically represented in the *Resource Tree* of the IPS Management Console as a hierarchical tree structure. Resources in the Network Security Manager are represented as nodes on the Resource Tree.

When a user logs into the Console, he will be presented with only those nodes that have been delegated to his particular domain or sub-domain. In addition, he will only be able to see alerts that relate directly to the nodes and resources under his administrative control.

The *System Health* screen provides a colour-coded (red, green or yellow) summary of the health of various system components, including the NSM, the NSP (IPS Sensor) and the database. Where the status is not green, hyperlinks provide instant access to the events that caused the problem, and system faults - like alerts - can be forwarded by severity to either an SNMP or Syslog server, sent to an administrator via e-mail or pager, or processed via a script. Once the events have been investigated and resolved, they can be acknowledged and the status will return to green.

The *Configure* screen provides a hierarchical *Resource Tree* down the left of the screen, and one or more tabbed configuration screens on the right. In addition to providing the ability to manage users and domains as mentioned earlier, the *Configure* screen also enables the administrator to configure system notification parameters (e-mail, pager, SNMP syslog or script), perform database backups and restores (these can be on demand or scheduled), acquire software and signature updates from the *Update Server* (on demand or scheduled), carry out system file maintenance operations (deleting old log file entries, for example, on demand or scheduled), manage LDAP and RADIUS authentication mechanisms, create custom signatures, and create, edit and deploy security policies.

New signatures and software patches are made available to customers over the Internet via the *Update Server*, which provides secure, fully automated, real-time signature updates without requiring any manual intervention. According to a user-configured schedule or via a manual process, the NSM polls the *Update Server*, and compares the file on the Update Server with what is already available in the NSM server to determine what needs to be downloaded.

Once it has received the update, the NSM then determines what signatures need to be pushed out to sensors based on the policy applied to the sensor. For example, a policy defined for a Windows environment will receive only updated signatures that apply to that environment.

The NSM compiles a specific update for each sensor and the update can then be pushed to sensors either manually, in an automated, real-time fashion or via automatic scheduled updates. Signature updates can be rolled out to all sensors at the click of a button and applied to each sensor in real time without requiring a reboot. It is also possible to roll back to a previous signature pack version just as easily.

State on existing connections is maintained during a signature update, and new signatures are even applied to subsequent packets of existing flows - this kind of continuous operation is very reassuring to administrators when deploying in-line devices.

Both sensor and policy configurations can be exported and imported between NSM servers. This would allow an administrator to operate both staging and production version of the NSM, and easily move configuration information between them.

2.4.2 POLICY MANAGEMENT

McAfee supplies a set of four pre-configured policies for immediate application: *Default IDS*, *Default In-Line IPS*, *All Inclusive With Audit* and *All Inclusive Without Audit*. The built-in policies are available in the *Policy Editor*, and should be considered as good starting points, designed to help get the system up and running quickly. The *All Inclusive without Audit* policy was the policy used during testing.

Any of the default scenarios can be applied and used as they stand, or they can be cloned (pre-defined policies cannot be edited directly) and modified in order to apply custom policies. The McAfee *Default In-Line IPS* policy, applied automatically when the first sensor is added, enables the administrator to begin protecting the network immediately with the most wide-ranging policy, but excluding possible “noisy” signatures.

All Network Security Platform (IPS) policies are rule-based - each rule in the set is either an *include* rule or an *exclude* rule, and determines what signature or group of signatures will be incorporated into the policy. An include rule usually starts a rule set and consists of a set of parameters that encompass a broad range of well-known attacks for detection.

More than one include rule can be applied if it is required to be specific about the rules included in a policy (specifying the inclusion of only HTTP and FTP rules, for example). One or more exclude rules can then be applied to remove elements from the include rules in order to focus the policy’s rule set further.

For example, if a sensor is operating in-line in front of a DMZ with only IIS Web servers, the administrator might include all HTTP signatures, and then exclude all non-IIS signatures and finally exclude all those signatures with a benign trigger probability greater than “*Low*”. That way, only the relevant signatures are

applied, and the administrator can be reasonably sure that false positives will not cause a self-inflicted Denial of Service condition.

The application of a rule set to a policy determines which attacks are included and displayed in the Policy Editor. Each attack has a number of default settings, including:

- *Whether the attack is enabled or disabled*
- *Attack severity*
- *Logging behavior - the sensor can log the entire packet or a specific number of bytes, and in addition can log 256 bytes of application data prior to the attack*
- *Duration of logging - the sensor can capture the attack packet only, capture a specified number of packets, for a specified length of time, capture the entire flow, or deliver forensic logging where all subsequent communication between attacker and victim can be logged.*
- *Sensor actions - send TCP reset (to source, destination or both), send "ICMP Host Not Reachable" packet to source, reconfigure the firewall, drop attack packet and all subsequent packets for that flow.*
- *Alert filter - enables the administrator to suppress certain alerts from or to specific IP addresses or a range of addresses*
- *Notifications - e-mail, pager or script notifications from the NSM to the administrator*

Any of these can be changed as required, and a useful bulk-edit capability allows the administrator to select multiple attacks (using standard Windows selection capabilities via the SHIFT and CTRL keys) and apply changes to an entire group of signatures in a single operation.

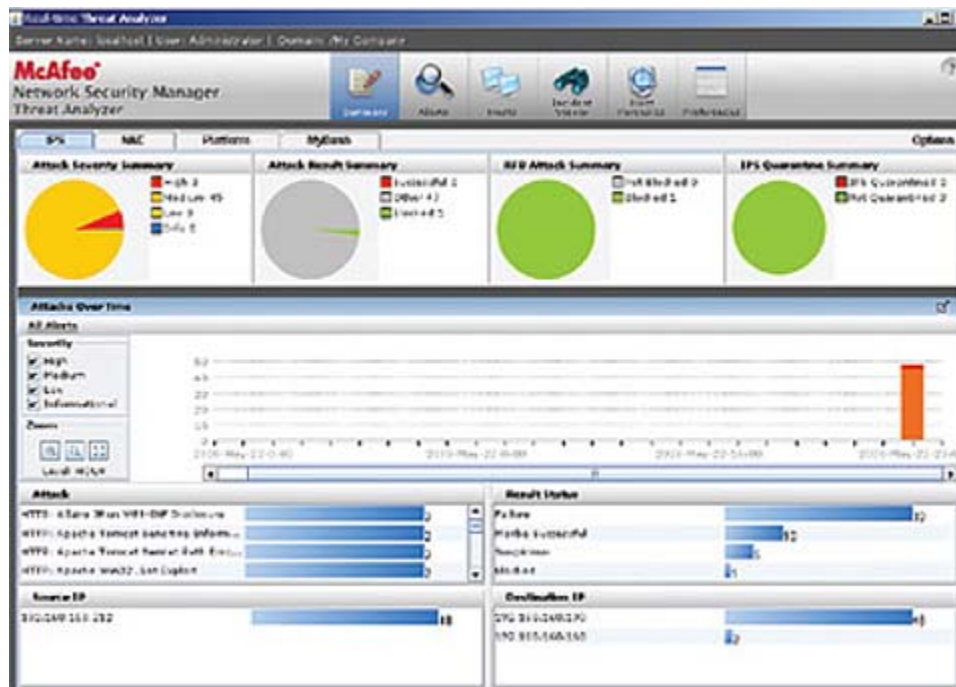
2.4.3 ALERT HANDLING

Alerts can be viewed via the real-time alert viewer or the historical alert viewer, both of which can be launched via drop-down menus on the console tool bar.

When an alert is first raised, it appears in the NSM Console in an *unacknowledged* state, and remains in that state until the administrator either acknowledges or deletes it. Acknowledging alerts dismisses them from the real-time views, after which they display only in the historical view in the alert viewer and in reports.

Alerts are backed up to the database and archived in order of occurrence, whereas deleted alerts are removed from the database altogether. Using the historical alert viewer, it is possible to return an *acknowledged* alert back to an *unacknowledged* state or un-delete an alert, providing it has not been removed permanently by the file maintenance utility.

The historical viewer sets the filter to retrieve information for both acknowledged and unacknowledged alerts written to the database within a specified time frame. It is also possible to apply other constraints - such as protocol or attack type - when launching the historical viewer. Older data can be archived to separate files based on date range, and can be restored to any NSM server for further analysis at a later date.



The alert viewer provides a useful graphical summary screen which contains a number of graphs of total alerts over time, top attacks, top source IPs, top destination IPs, and so on. In the detailed view (accessible via a toolbar button), alerts can be shown as a simple chronological list or grouped together in various ways via a drop-down menu (by protocol, attack name, severity, IP address, and so on).

The real power of the alert viewer lies in the drill-down capabilities. Right-clicking or double-clicking on any column in a graph in the summary view will cause the remaining graphs to be redrawn with new data. The detail view then contains only those alerts which correspond to the more focused graphical summary view.

There are many different ways of slicing and dicing data in order to drill down from very high level summaries to very low level detail. It is also possible to save selected windows in PDF or CSV format to create quick reports. Overall the system works extremely well.

Double clicking on an individual alert entry brings up the *Exploit Alert Detail* windows for that entry with multiple tabbed windows. The *Exploit* tab provides information on the *alert*, including the attack name, sensor ID, interface name, severity, time, domain, alert ID and a link to a detailed description of the attack.

Right-clicking on an alert entry provides a *context-sensitive* menu from where the administrator can acknowledge and alert, create an incident, look up all related alerts, or create/edit the associated attack response at a policy or global level.

The *Alert* tab provides details specific to the attack type, which could be an *exploit*, a *DoS attack*, a *port scan*, and so on. This tab includes information such as source and destination IP address, network protocol, application protocol, threshold values, and target ports. The *Response* tab provides response information,

where the administrator can examine the exploit packet, including the 256 bytes immediately preceding it or even the entire flow if those logging options were enabled for that particular signature.

A packet log is created by a Network Security Platform (IPS) sensor capturing the network traffic of and around an offending transmission. If logging was enabled for specific exploit signatures (the *Configured Response* tab provides details on exactly which responses were triggered by the alert, and thus provides an indication to the administrator of exactly what logging data is available), the appropriate packet logs are saved in library packet capture (libpcap) format, and stored in the NSM database. *WireShark (Ethereal)* is used as the packet viewer.

The final tab in the detailed alert view window is the *Signature Description*, which provides some information on exactly what caused the packet to trigger this particular alert. This is useful given that it is not possible to examine the actual signatures within the Network Security Platform (IPS) system.

Once an alert has been examined and investigated, it can be acknowledged. At that point it is removed from the various statistical/summary views, and is subsequently only retrieved from the database for searches in the historical alert viewer and IPS reports.

An alert can also be saved as an *Evidence Report*. This opens a complete view of a selected alert row in a separate window, and provides the option to save the alert information, including a packet log (if available), in a zip file which can be saved or passed to others for forensic analysis.

For more extensive forensic analysis, Network Security Platform (IPS) provides the concept of “*Incidents*”. An Incident is a collection of related alerts which are created manually by the administrator, who can select and group together a collection of related alerts from the alert viewer.

Defining an incident enables the administrator to build a file for research, to be used in an investigation, or any other assortment of forensic analysis uses. The *Incident Viewer* displays incident statistics, provides a comments area for case management purposes, and enables deletion of incidents, and basic workflow capabilities allow each incident to be assigned to, and annotated by, a number of different personnel.

Assigning a responsible party is useful for quick recognition upon future opening of the incident and is very helpful in multiple administrator environments where more than one person will perform tasks on collected data.

In all respects, the alert handling capabilities are extremely comprehensive and powerful and yet relatively easy to use once the interface has been mastered. The drill-down and correlation (incident) capabilities make it very easy for administrators to focus on the relevant detail.

2.4.4 REPORTING AND ANALYSIS

Whilst the alert *viewer* provides both real-time monitoring and interactive forensic analysis capabilities (as well as the ability to save selected windows in PDF/CSV format to create quick reports), the *Report Generator* offers the administrator the opportunity to create more comprehensive, and higher-level summary reports both in text and graphical format.

The reports provide summary information on the alerts generated from the installed sensors. The generated alert information can include source and destination IP of the attack, time when attack occurred, sensor that

detected the attack, and so forth. The multiple reports in this category provide various, concentrated views according to the specific parameters of each report, and each report lists alerts from most to least common detected.

There are six options in the *IPS Reports* menu, including four pre-defined reports, a user-defined reports option, and a template management capability:

- **Executive Summary Report** - Provides a summary view of selected alert data presented in a variety of tables, graphs, and charts.
- **Top N Report** - Lists a count of alerts in order of frequency for one of four defining categories: attack type, source IP, destination IP, or source/destination IP pair.
- **User-Defined Report** - Presents alerts based on a variety of user-defined filters including interface, IP address, port number, application protocol, and direction of alert.
- **Reconnaissance Report** - Provides a summary of all reconnaissance alerts (scans, sweeps, probes) detected during a specified time frame.
- **Trend Analysis Report** - Presents alert data based on common trends per specified frequency (e.g., number of high severity alerts per hour for one day).
- **Report Templates** - Enables the administrator to create custom IPS report templates that can be run on-demand, as well as manage the report templates which were created for Scheduled Reports. IPS report templates simplify the process of generating a frequently used report by enabling the administrator to create a template for a report, and simply return to this action to generate the report based on the saved settings at any given point in the future.

Configuration reports provide information on the settings applied using the *System Configuration* tool. Reports can be generated to view admin-related information such as the current software and signature versions, the status of a sensor, or policy settings.

The *Scheduled Reports* options simplify the reporting process by automating the report generation procedure. Scheduled reports can be generated and e-mailed on a daily or weekly basis.

All reports can be viewed in either HTML or PDF format. The *Top N Report* can also be viewed in bar graph or pie chart format.

3 TEST RESULTS SCORECARD

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

Result	Test ID	Description	Comment
	5.1	Detection Engine	
PASS	5.1.1	System Exposure	99.4%
PASS	5.1.2	Service Exposure	99.0%
PASS	5.1.3	System or Service Fault	100%
	5.2	Threat Vectors	
PASS	5.2.1	Attacker Initiated	99.8%
PASS	5.2.2	Target Initiated	98.6%
PASS	5.2.3	Network	99.4%
N/A	5.2.4	Local	Not for NIPS
	5.3	Evasion	
PASS	5.3.1	Evasion	100%
	5.4	Packet Fragmentation	
PASS	5.4.1	Ordered 8 byte fragments	100%
PASS	5.4.2	Ordered 24 byte fragments	100%
PASS	5.4.3	Out of order 8 byte fragments	100%
PASS	5.4.4	Ordered 8 byte fragments, duplicate last packet	100%
PASS	5.4.5	Out of order 8 byte fragments, duplicate last packet	100%
PASS	5.4.6	Ordered 8 byte fragments, reorder fragments in reverse	100%
PASS	5.4.7	Ordered 16 byte frags, fragment overlap (favor new)	100%
PASS	5.4.8	Ordered 16 byte frags, fragment overlap (favor old)	100%
PASS	5.4.9	Out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery	100%
	5.5	Stream Segmentation	
PASS	5.5.1	Ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums	100%
PASS	5.5.2	Ordered 1 byte segments, interleaved duplicate segments with null TCP control flags	100%
PASS	5.5.3	Ordered 1 byte segs, interleaved duplicate segments with requests to resync sequence numbers mid-stream	100%
PASS	5.5.4	Ordered 1 byte segments, duplicate last packet	100%
PASS	5.5.5	Ordered 2 byte segments, segment overlap (favor new)	100%

Result	Test ID	Description	Comment
PASS	5.5.6	Ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers	100%
PASS	5.5.7	Out of order 1 byte segments	100%
PASS	5.5.8	Out of order 1 byte segments, interleaved duplicate segments with faked retransmits	100%
PASS	5.5.9	Ordered 1 byte segments, segment overlap (favor new)	100%
PASS	5.5.10	Out of order 1 byte segs, PAWS elimination (interleaved dup segs with older TCP timestamp options)	100%
PASS	5.5.11	Ordered 16 byte segs, seg overlap (favor new (Unix))	100%
	5.6	RPC Fragmentation	
PASS	5.6.1	One-byte fragmentation (ONC)	100%
PASS	5.6.2	Two-byte fragmentation (ONC)	100%
PASS	5.6.3	All fragments, including Last Fragment (LF) will be sent in one TCP segment (ONC)	100%
PASS	5.6.4	All frags except Last Fragment (LF) will be sent in one TCP segment. LF will be sent in separate TCP seg (ONC)	100%
PASS	5.6.5	One RPC fragment will be sent per TCP segment (ONC)	100%
PASS	5.6.6	One LF split over more than one TCP segment. In this case no RPC fragmentation is performed (ONC)	100%
PASS	5.6.7	Canvas Reference Implementation Level 1 (MS)	100%
PASS	5.6.8	Canvas Reference Implementation Level 2 (MS)	100%
PASS	5.6.9	Canvas Reference Implementation Level 3 (MS)	100%
PASS	5.6.10	Canvas Reference Implementation Level 4 (MS)	100%
PASS	5.6.11	Canvas Reference Implementation Level 5 (MS)	100%
PASS	5.6.12	Canvas Reference Implementation Level 6 (MS)	100%
PASS	5.6.13	Canvas Reference Implementation Level 7 (MS)	100%
PASS	5.6.14	Canvas Reference Implementation Level 8 (MS)	100%
PASS	5.6.15	Canvas Reference Implementation Level 9 (MS)	100%
PASS	5.6.16	Canvas Reference Implementation Level 10 (MS)	100%
	5.7	URL Obfuscation	
PASS	5.7.1	URL encoding - Level 1 (minimal)	100%
PASS	5.7.2	URL encoding - Level 2	100%
PASS	5.7.3	URL encoding - Level 3	100%
PASS	5.7.4	URL encoding - Level 4	100%
PASS	5.7.5	URL encoding - Level 5	100%
PASS	5.7.6	URL encoding - Level 6	100%
PASS	5.7.7	URL encoding - Level 7	100%

Result	Test ID	Description	Comment
PASS	5.7.8	URL encoding - Level 8 (extreme)	100%
PASS	5.7.9	Premature URL ending	100%
PASS	5.7.10	Long URL	100%
PASS	5.7.11	Fake parameter	100%
PASS	5.7.12	TAB separation	100%
PASS	5.7.13	Case sensitivity	100%
PASS	5.7.14	Windows \ delimiter	100%
PASS	5.7.15	Session splicing	100%
	5.8	FTP Evasion	
PASS	5.8.1	Inserting spaces in FTP command lines	100%
PASS	5.8.2	Inserting non-text Telnet opcodes - Level 1 (minimal)	100%
PASS	5.8.3	Inserting non-text Telnet opcodes - Level 2	100%
PASS	5.8.4	Inserting non-text Telnet opcodes - Level 3	100%
PASS	5.8.5	Inserting non-text Telnet opcodes - Level 4	100%
PASS	5.8.6	Inserting non-text Telnet opcodes - Level 5	100%
PASS	5.8.7	Inserting non-text Telnet opcodes - Level 6	100%
PASS	5.8.8	Inserting non-text Telnet opcodes - Level 7	100%
PASS	5.8.9	Inserting non-text Telnet opcodes - Level 8 (extreme)	100%
	6	NIPS Performance	
	6.1	Raw Packet Processing Performance (UDP Traffic)	
PASS	6.1.1	128 Byte Packets	10.4 Gbps
PASS	6.1.2	256 Byte Packets	19.2 Gbps
PASS	6.1.3	512 Byte Packets	20 Gbps
PASS	6.1.4	1024 Byte Packets	20 Gbps
PASS	6.1.5	1514 Byte Packets	20 Gbps
	6.2	Maximum Capacity	
PASS	6.2.1	Theoretical Max. Concurrent TCP Connections	5,509,000
PASS	6.2.2	Theoretical Max. Concurrent TCP Connections w/Data	5,026,600
PASS	6.2.3	Stateful Protection at Max Concurrent Connections	100%
PASS	6.2.4	Maximum TCP Connections Per Second	314,000
PASS	6.2.5	Maximum HTTP Connections Per Second	145,000
PASS	6.2.6	Maximum HTTP Transactions Per Second	275,000
	6.3	Behavior Of The State Engine Under Load	
PASS	6.3.1	Attack Detection/Blocking - Normal Load	100%
PASS	6.3.2	State Preservation - Normal Load	100%
PASS	6.3.3	Pass Legitimate Traffic - Normal Load	100%

Result	Test ID	Description	Comment
PASS	6.3.4	Attack Detection/Blocking - Maximum Exceeded	100%
PASS	6.3.5	State Preservation - Maximum Exceeded	100%
PASS	6.3.6	Pass Legitimate Traffic - Maximum Exceeded	100%
	6.4	HTTP Capacity With No Transaction Delays	
PASS	6.4.1	2,500 Connections Per Second – 44Kbyte Response	39,250 CPS
PASS	6.4.2	5,000 Connections Per Second – 21Kbyte Response	57,000 CPS
PASS	6.4.3	10,000 Connections Per Second – 10Kbyte Response	77,000 CPS
PASS	6.4.4	20,000 Connections Per Second – 4.5Kbyte Response	89,000 CPS
PASS	6.4.5	40,000 Connections Per Second – 1.7Kbyte Response	116,000 CPS
	6.5	HTTP Capacity With Transaction Delays	
	6.5.1	21 Kbyte Response With Delay	54000 CPS
	6.5.2	10 Kbyte Response With Delay	73000 CPS
	6.6	“Real World” Traffic	
PASS	6.6.1	“Real World” Protocol Mix (Perimeter)	11.5 Gbps
PASS	6.6.2	“Real World” Protocol Mix (Core)	10.1 Gbps
	6.7	Latency	
PASS	6.7.1	Latency 128 256 512 1024 1514	51 μ s 88 μ s 191 μ s 311 μ s 384 μ s
	6.8	User Response Times	
PASS	6.8.1	Application Response With No Background Traffic (21Kbyte Response) <ul style="list-style-type: none"> • TCP Average Time to Open • TCP Average Time to Response Packet • TCP Average Time to Close • Application Average Response Time: HTTP • Application Average Response Time: SMTP • Application Average Response Time: DNS 	0.65 ms 0.57 ms 0.92 ms 32 ms 11 ms 51 ms
	7	Stability & Reliability	
PASS	7.1.1	Blocking Under Extended Attack	100%
PASS	7.1.2	Passing Legitimate Traffic Under Extended Attack	100%
PASS	7.1.3	Protocol Fuzzing	100%
PASS	7.1.4	Protocol Mutation	100%
PASS	7.1.5	Policy Push	100%
PASS	7.1.6	Power Fail	100%
YES	7.1.7	Redundancy	Multiple Power, Fans, etc.

Result	Test ID	Description	Comment
OPTION	7.1.8	Fail Open (Power Fail/Reboot)	External Bypass
N/A	7.1.9	Fail Open (Resource Issues)	N/A
YES	7.1.10	Fail Closed (Power Fail/Reboot)	Default
PASS	7.1.11	Fail Closed (Resource Issues)	Default
YES	7.1.12	High Availability (HA) Option (Stateful)	Default
N/A	7.1.13	High Availability (HA) Option (Non-stateful)	N/A
PASS	7.1.14	Persistence Of Data	Default
PASS	7.1.15	IPV6	Option in the network settings
	8	Management and Configuration	
	8.1	Management Port	
PASS	8.1.1	Open Ports Required	161, 22
PASS	8.1.2	Open Ports Detected	161, 22
PASS	8.1.3	Protocol Fuzzing	100%
PASS	8.1.4	Protocol Fuzzing Detection	100%
	8.2	Management & Configuration - General	
PASS	8.2.1	Transparent Mode	Default
PASS	8.2.2	Management Port	Dedicated Interface
PASS	8.2.3	Management Protocol	Secure
PASS	8.2.4	Authentication	Secure
PASS	8.2.5	Enterprise Authentication	Configurable
PASS	8.2.6	Direct NIPS Management (Optional)	Command Line
PASS	8.2.7	Centralized NIPS Management	Available via Network Security Central Manager
PASS	8.2.8	Pass-Through Mode (Optional)	Activated via CLI
PASS	8.2.9	Signature Update	Automatic or Manual
PASS	8.2.10	Secure NIPS Registration	Simple
PASS	8.2.11	Documentation	Comprehensive (See website)
	8.3	Management & Configuration – Policy	
PASS	8.3.1	NIPS Configuration	Network Security Central Manager and Network Security Manager able to manage multiple devices
PASS	8.3.2	Policy Definition	Flexible and Powerful

Result	Test ID	Description	Comment
PASS	8.3.3	Recommended Settings	Easy to find and deploy
PASS	8.3.4	Custom Attack Signatures	Possible via Network Security Manager
PASS	8.3.5	Bulk Operations	Possible via Network Security Central Manager and Network Security Manager
PASS	8.3.6	Granularity	Excellent
PASS	8.3.7	Policy Association	Excellent and simple
PASS	8.3.8	Inheritance	Default behavior
PASS	8.3.9	Virtualization	Possible via setting policy per Interface pair
PASS	8.3.10	Policy Deployment	Simple via Network Security Manager or Central Manager
PASS	8.3.11	Policy Auditing	Detailed reports citing which changes which made and by whom
PASS	8.3.12	Policy Version Control	Automated and simple to use
	8.4	Management & Configuration - Alert Handling	
PASS	8.4.1	Required Log Events	Full Capabilities
PASS	8.4.2	Log Location (Optional)	Configurable
PASS	8.4.3	Communication Interruption	Configurable
PASS	8.4.4	Log Flooding	Handled Properly
PASS	8.4.5	Alerts	Configurable
PASS	8.4.6	Alert Accuracy	Excellent detail
PASS	8.4.7	Centralized Alerts	Excellent detail
PASS	8.4.8	Alert Delivery MechanNSM	Email, pager, SNMP
PASS	8.4.9	Alert Actions (Mandatory)	Full Capabilities
PASS	8.4.10	Alert Actions (Optional)	Send notice to Pager
PASS	8.4.11	Forensic Analysis	Full details available
PASS	8.4.12	Summarize Alerts	Available via UI
PASS	8.4.13	View Alert Detail	Available via UI

Result	Test ID	Description	Comment
PASS	8.4.14	View Related Policy	Single click from Alert
PASS	8.4.15	View Packet Contents (Optional)	Available in drill-down
PASS	8.4.16	Alert Suppression	Configurable
PASS	8.4.17	Correlation (Automatic)	Provided via Dashboard in UI
PASS	8.4.18	Correlation (Manual)	Provided via Dashboard in UI
PASS	8.4.19	Incident Workflow	Possible with plug-ins to other products
	8.5	Management & Configuration – Reporting	
PASS	8.5.1	Centralized Reports	Available via UI
PASS	8.5.2	Top Attacks	Displayed in UI Dashboard
PASS	8.5.3	Top Sources	Displayed in UI Dashboard
PASS	8.5.4	Top Targets	Displayed in UI Dashboard
PASS	8.5.5	Top Services	Displayed in UI Dashboard
PASS	8.5.6	Top Protocols	Displayed in UI Dashboard
PASS	8.5.7	Custom Reports	Available via UI
PASS	8.5.8	Saved Reports	Available via UI
PASS	8.5.9	Scheduled Reports	Available via UI
PASS	8.5.10	Log File Maintenance	Available via UI

4 THE PRODUCT UNDER TEST – MCAFEE M-8000

4.1 M-8000 DUAL SENSOR IPS

The McAfee M-8000 sensor is a content processing appliance built to inspect, detect and block intrusions, misuse, and distributed denial of service (DDoS) attacks. When deployed at key network access points, an M-8000 sensor provides real-time traffic monitoring to detect and responds to malicious activity as configured by the administrator

McAfee M-8000 Network Security Platform provides protection against threats and attacks including:

- Zero-day attacks, cyber-attacks, and malware
- Spyware, phishing, and other unwanted programs
- Voice over IP (VoIP) threats and vulnerabilities
- Denial of service (DoS), distributed DoS (DDoS), and SYN flood attacks
- Encrypted attacks, worms, Trojans, and evasions
- Instant messaging and peer-to-peer applications

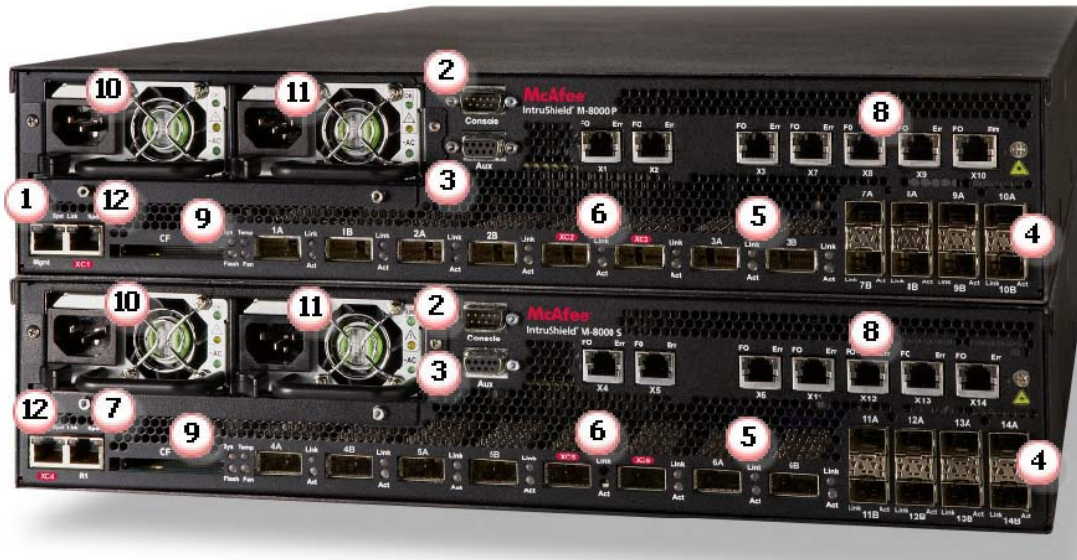
In addition to threat prevention, the following features are also included in the M-8000 Network Security Platform:

- Adaptive rate limiting—Real-time, adaptive protocol rate shaping controls network bandwidth while blocking unwanted and risky applications
- Built-in host quarantine—Real-time quarantine protection provides automated host quarantine

4.2 M-8000 HARDWARE

The McAfee Network Security Platform M-8000 sensor consists of two (2) x 2RU units and is equipped with:

- Six (6) x 10 Gigabit INLINE segments = twelve (12) x 10 Gigabit SPAN ports
- Eight (8) x 1 Gigabit INLINE segments = sixteen (16) x 1 Gigabit SPAN ports
- Hot-swappable SFP/XFP modules
- Dual Power Supplies
- Fan units (that are field replaceable)
- One (1) Response port
- One (1) 10/100/1000 Management port



Name	Description
1	Management port (on M-8000 P only)
2	Console port
3	Auxiliary port Two RS-232C Auxiliary ports, which may be used to dial in remotely to set up and configure the sensor
4	SFP Gigabit Ethernet Monitoring ports Sixteen small form-factor pluggable (SFP) 1 Gigabit ports
5	XFP 10-Gigabit Ethernet Monitoring ports Twelve 10-Gigabit small form-factor pluggable (XFP) ports
6	XFP Interconnect ports Four 10-Gigabit small form-factor pluggable (XFP) Interconnect ports
7	Response port (on M-8000 S only)
8	Fail-Open Control ports Fourteen RJ-11 Fail-Open Control ports, designed for use the Optical Fail-Open Bypass kit.
9	External Compact Flash ports
10	Power Supply A Two Primary Power Supplies—A (included).
11	Power Supply B Two Power Supplies—B (optional, purchased separately).
12	10/100/1000 Interconnect ports

Twelve (12) 10-Gigabit small form-factor pluggable (XFP) ports, enable monitoring of twelve (12) SPAN ports, six full-duplex tapped segments, six in-line segments, or a combination. Sixteen small form-factor pluggable (SFP) 1-Gigabit ports enable monitoring of sixteen (16) SPAN ports, eight full-duplex tapped segments, eight in-line segments, or a combination. The Monitoring interfaces of the M-8000 work in stealth mode; they have no IP address and are not visible on the monitored segment. For Failover Mode, ports 4A and 4B are used to interconnect with a standby sensor.

There are four (4) 10-Gigabit small form-factor pluggable (XFP) and two RJ-45 10/100/1000 Interconnect ports for communication between the primary sensor and the secondary sensor. The Interconnect interfaces of the M-8000 work in stealth mode; they have no IP address and are not visible on the monitored segment.

The M-8000 leverages an external bypass kit for Fail-Open capability and provides fourteen RJ-11 Fail-Open Control ports, designed for use the Optical Fail-Open Bypass kit. The ports are marked X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, and X14 and are used in conjunction with ports 1A/1B, 2A/2B, 3A/3B, 4A/4B, 5A/5B, 6A/6B, 7A/7B, 8A/8B, 9A/9B, 10A/10B, 11A/11B, 12A/12B, 13A/13B, and 14A/14B, respectively. There are two (2) External Compact Flash ports that are used to control optional the fail-open hardware (Gigabit Optical Fail-Open Bypass Kit) as well as for troubleshooting situations where the sensor's internal flash is corrupted and you must reboot the sensor via the external compact flash.

The dual sensor ships with two (2) Primary Power Supplies—A (included). The power supply uses a standard IEC port (IEC320-C13, and McAfee provides a standard, 2m NEMA 5-15P (US) power cable (3 wire). International customers must procure a country-appropriate power cable. There is also capacity for two (2) optional secondary power supplies. Power supply B is a hot-swappable, redundant power supply that uses a standard IEC320-C13 port. McAfee provides a standard 2m NEMA 5-15P (US) power cable (3 wire)

4.3 McAfee Network Security Manager

The McAfee Network Security Manager (NSM) is offered as an appliance or software that delivers centralized, web-based management of Network Security Platform Sensors. The console enables management, configuration, and monitoring of NSP appliances. The web-based management interface supports single or multiple device deployments.

McAfee Network Security Manager includes:

- Ability to control configurations and policies for multiple IPS sensors
- Pre-defined policies
- Real-time viewer of installed Network Security Platform sensors

There are three flavors of NSM:

- Global Manager—more than six sensors
- Manager—up to six sensors
- Manager Starter—up to two sensors

5 TEST METHODOLOGY – SECURITY EFFECTIVENESS

This section verifies that the DUT is capable of detecting and blocking a wide range of common exploits accurately, while remaining resistant to false positives. All tests are performed initially with no background network load. The tests are then repeated under varying levels and mixes of background traffic to ensure that the results do not vary when handling normal network traffic.

The latest signature pack is acquired from the vendor, and the DUT is deployed with the default security policy or recommended settings based on the target Appropriate Usage environment. NSS Labs considers it unacceptable for a product of this nature to be sold without a default policy and/or recommended settings, or without consultancy included to create a policy specific to the target environment. No custom signatures are permitted in the testing - all signatures used must be available to the general public at the time of testing.

Although IDS operate in detection only mode, a NIPS is required to block and log exploit attempts and hostile traffic. However, Denial of Service attacks are left to the dedicated NSS Network Intrusion Prevention System testing track.

5.1 DETECTION ENGINE

While it is not possible to validate the entire signature set of any DUT, the NSS Labs testing provides a demonstration of effectiveness for the DUT to protect vulnerable assets from targeted threats and exploitation. This asset/target and threat based approach forms the basis from which NIPS security effectiveness is measured. (see NSS Labs' whitepaper *Intrusion Prevention Security Effectiveness*).

The NSS threat and attack suite contains thousands of publically available exploits (including multiple variants of each exploit) from which groups of exploits are carefully selected to test based on *Appropriate Usage*. Each exploit has been validated to impact the target vulnerable host(s). Based on the impact of the threat against the target the following metrics are reported:

5.1.1 SYSTEM EXPOSURE

Attacks resulting in remote system compromise and the ability of the attacker to execute arbitrary system level commands. Most exploits in this class that are “weaponized” will provide the attacker with a fully interactive remote shell on the target client or server.

5.1.2 SERVICE EXPOSURE

Attacks resulting 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 the 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 go from the service level to the system level.

5.1.3 SYSTEM OR SERVICE FAULT

Attacks resulting 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. However the resulting impact to the business could be severe given that the attacker could crash the protected system or service.

5.2 THREAT VECTOR

Threats and exploits can be initiated by either the target or the attacker targeting either local or remote vulnerabilities. As a result, threats and exploits are categorized into the following matrix:

	Network	Local
Attacker	RPC Exploit	Root Kit
Target	Browser Exploit	Trojan

*Example exploits included above for reference purposes.

5.2.1 ATTACKER INITIATED

The threat/exploit is executed by the attacker remotely against a vulnerable application and/or operating system.

5.2.2 TARGET INITIATED

The threat/exploit is initiated by the vulnerable target. The attacker has little or no control as to when the target user or application will execute the threat.

5.2.3 NETWORK

Threat/exploits that are initiated as a result of network communication.

5.2.4 LOCAL

Local execution that requires existing access to the target (not applicable to NIPS).

Protective ratings are reported in raw percentages of mitigated attacks and their resulting impact: *System, Service, Fault, Reconnaissance*. Although a system or service exploit may be partially mitigated by the DUT, the service could have crashed because of the residual communications resulting in a Fault impact on the service or OS. (e.g. The exploit did not produce a remote shell but crashed the target instead.)

5.3 EVASION

This section verifies that the DUT is capable of detecting and blocking basic exploits when subjected to varying common evasion techniques.

5.3.1 UNMODIFIED EXPLOIT VALIDATION

A number of common exploits are executed across the DUT to ensure that they are detected in their unmodified state. These will be chosen from a suite of older/common basic exploits for which NSS is certain that all vendors will have signatures. None of the exploits that were used in Section 5.1 will be used as evasion baselines. This ensures that vendors are not provided with any information on the content of any part of the main NSS exploit library in advance of the test.

5.4 PACKET FRAGMENTATION

These tests determine the effectiveness of the fragment reassembly mechanism of the DUT.

- 5.4.1 ORDERED 8 BYTE FRAGMENTS**
- 5.4.2 ORDERED 24 BYTE FRAGMENTS**
- 5.4.3 OUT OF ORDER 8 BYTE FRAGMENTS**
- 5.4.4 ORDERED 8 BYTE FRAGMENTS, DUPLICATE LAST PACKET**
- 5.4.5 OUT OF ORDER 8 BYTE FRAGMENTS, DUPLICATE LAST PACKET**
- 5.4.6 ORDERED 8 BYTE FRAGMENTS, REORDER FRAGMENTS IN REVERSE**
- 5.4.7 ORDERED 16 BYTE FRAGMENTS, FRAGMENT OVERLAP (FAVOR NEW)**
- 5.4.8 ORDERED 16 BYTE FRAGMENTS, FRAGMENT OVERLAP (FAVOR OLD)**
- 5.4.9 OUT OF ORDER 8 BYTE FRAGMENTS, INTERLEAVED DUPLICATE PACKETS SCHEDULED FOR LATER DELIVERY**

It is a requirement of the test that DUT submitted should have all IP fragmentation reassembly options enabled by default in the shipping product.

5.5 STREAM SEGMENTATION

These tests determine the effectiveness of the stream reassembly mechanism of the DUT.

- 5.5.1 ORDERED 1 BYTE SEGMENTS, INTERLEAVED DUPLICATE SEGMENTS WITH INVALID TCP CHECKSUMS**
- 5.5.2 ORDERED 1 BYTE SEGMENTS, INTERLEAVED DUPLICATE SEGMENTS WITH NULL TCP CONTROL FLAGS**
- 5.5.3 ORDERED 1 BYTE SEGMENTS, INTERLEAVED DUPLICATE SEGMENTS WITH REQUESTS TO RESYNC SEQUENCE NUMBERS MID-STREAM**
- 5.5.4 ORDERED 1 BYTE SEGMENTS, DUPLICATE LAST PACKET**
- 5.5.5 ORDERED 2 BYTE SEGMENTS, SEGMENT OVERLAP (FAVOR NEW)**
- 5.5.6 ORDERED 1 BYTE SEGMENTS, INTERLEAVED DUPLICATE SEGMENTS WITH OUT-OF-WINDOW SEQUENCE NUMBERS**
- 5.5.7 OUT OF ORDER 1 BYTE SEGMENTS**
- 5.5.8 OUT OF ORDER 1 BYTE SEGMENTS, INTERLEAVED DUPLICATE SEGMENTS WITH FAKED RETRANSMITS**
- 5.5.9 ORDERED 1 BYTE SEGMENTS, SEGMENT OVERLAP (FAVOR NEW)**
- 5.5.10 OUT OF ORDER 1 BYTE SEGMENTS, PAWS ELIMINATION (INTERLEAVED DUP SEGS WITH OLDER TCP TIMESTAMP OPTIONS)**
- 5.5.11 ORDERED 16 BYTE SEGMENTS, SEGMENT OVERLAP (FAVOR NEW (UNIX))**

It is a requirement of the test that DUT submitted should have all TCP stream reassembly options enabled by default in the shipping product.

5.6 RPC FRAGMENTATION

Both Sun/ONC RPC and MS-RPC allow the sending application to fragment requests, and all MS-RPC services have a built-in fragmentation reassembly mechanism.

An attacker can transmit the BIND followed by a single request fragmented over a hundred actual requests with small fragments of the malicious payload. Alternatively, the attacker could transmit both the BIND and request fragments in one large TCP segment, thus foiling any signatures which use a simple size check.

Immunitysec's CANVAS test tool combines large writes with many tiny MS-RPC fragments, and provides up to ten levels of fragmentation. These tests determine the effectiveness of the RPC reassembly mechanism of the DUT:

5.6.1 ONE-BYTE FRAGMENTATION (ONC)

5.6.2 TWO-BYTE FRAGMENTATION (ONC)

5.6.3 ALL FRAGMENTS, INCLUDING LAST FRAGMENT (LF) WILL BE SENT IN ONE TCP SEGMENT (ONC)

5.6.4 ALL FRAGMENTS EXCEPT LAST FRAGMENT (LF) WILL BE SENT IN ONE TCP SEGMENT. LF WILL BE SENT IN SEPARATE TCP SEGMENT (ONC)

5.6.5 ONE RPC FRAGMENT WILL BE SENT PER TCP SEGMENT (ONC)

5.6.6 ONE LF SPLIT OVER MORE THAN ONE TCP SEGMENT. IN THIS CASE NO RPC FRAGMENTATION IS PERFORMED (ONC)

5.6.7 CANVAS REFERENCE IMPLEMENTATION LEVEL 1 (MS)

5.6.8 CANVAS REFERENCE IMPLEMENTATION LEVEL 2 (MS)

5.6.9 CANVAS REFERENCE IMPLEMENTATION LEVEL 3 (MS)

5.6.10 CANVAS REFERENCE IMPLEMENTATION LEVEL 4 (MS)

5.6.11 CANVAS REFERENCE IMPLEMENTATION LEVEL 5 (MS)

5.6.12 CANVAS REFERENCE IMPLEMENTATION LEVEL 6 (MS)

5.6.13 CANVAS REFERENCE IMPLEMENTATION LEVEL 7 (MS)

5.6.14 CANVAS REFERENCE IMPLEMENTATION LEVEL 8 (MS)

5.6.15 CANVAS REFERENCE IMPLEMENTATION LEVEL 9 (MS)

5.6.16 CANVAS REFERENCE IMPLEMENTATION LEVEL 10 (MS)

5.7 URL OBFUSCATION

Random URL encoding techniques are employed to transform simple URLs which are often used in pattern-matching signatures to apparently meaningless strings of escape sequences and expanded path characters using a combination of the following techniques:

- Escape encoding (% encoding)
- Microsoft %u encoding
- Path character transformations and expansions (./, //, \)

These techniques are combined in various ways for each URL tested, ranging from minimal transformation, to extreme (every character transformed). All transformed URLs are verified to ensure they still function as expected after transformation.

5.7.1 URL ENCODING - LEVEL 1 (MINIMAL)

5.7.2 URL ENCODING - LEVEL 2

5.7.3 URL ENCODING - LEVEL 3

5.7.4 URL ENCODING - LEVEL 4

5.7.5 URL ENCODING - LEVEL 5

5.7.6 URL ENCODING - LEVEL 6

5.7.7 URL ENCODING - LEVEL 7

5.7.8 URL ENCODING - LEVEL 8 (EXTREME)

5.7.9 PREMATURE URL ENDING

5.7.10 LONG URL

5.7.11 FAKE PARAMETER

5.7.12 TAB SEPARATION

5.7.13 CASE SENSITIVITY

5.7.14 WINDOWS \ DELIMITER

5.7.15 SESSION SPLICING

5.8 FTP EVASION

When attempting FTP exploits, it is possible to evade some IDS/NIPS products by inserting additional spaces and telnet control sequences in FTP commands.

These tests insert a range of valid telnet control sequences that can be parsed and handled by IIS FTP server and wu-ftpd, and which also conform to Section 2.3 of RFC 959. Control opcodes are inserted at random, ranging from minimal insertion (only one pair of opcodes), to extreme (opcodes between every character in the FTP command):

5.8.1 INSERTING SPACES IN FTP COMMAND LINES

5.8.2 INSERTING NON-TEXT TELNET OPCODES - LEVEL 1 (MINIMAL)

5.8.3 INSERTING NON-TEXT TELNET OPCODES - LEVEL 2

5.8.4 INSERTING NON-TEXT TELNET OPCODES - LEVEL 3

5.8.5 INSERTING NON-TEXT TELNET OPCODES - LEVEL 4

5.8.6 INSERTING NON-TEXT TELNET OPCODES - LEVEL 5

5.8.7 INSERTING NON-TEXT TELNET OPCODES - LEVEL 6

5.8.8 INSERTING NON-TEXT TELNET OPCODES - LEVEL 7

5.8.9 INSERTING NON-TEXT TELNET OPCODES - LEVEL 8 (EXTREME)

6 TEST METHODOLOGY – NIPS PERFORMANCE

This section measures the performance of the DUT using various traffic conditions that provide metrics for real world performance. Individual implementations will vary based on usage, however these quantitative metrics provide a gauge as to whether a particular DUT is appropriate for a given environment.

The latest signature pack or rule set is acquired from the vendor, and sensors are deployed with the default/recommended settings applied as used for the *Security Effectiveness* testing. Each sensor is configured to detect and block suspicious traffic. The DUT should also be configured to block all traffic when resources are exhausted or when traffic cannot be analyzed for any reason. Any device which passes malicious traffic under the above conditions will fail.

Multiple separate 1Gbps connections will be made from the external to internal switches via the DUT, subject to a minimum of one in-line port pair per Gigabit of throughput. Thus an 8Gbps device with only four port pairs will be limited to 4Gbps. The minimum number of port pairs will be connected to support the claimed maximum bandwidth of the DUT. Thus an 8 Gbps device with ten port pairs will be deployed with eight 1Gbps connections.

Attacks are launched through the DUT against protected hosts with zero background traffic to ensure the DUT is capable of detecting the baseline attacks. Once that has been established, increasing levels of varying types of background traffic are generated through the DUT in order to determine the point at which the DUT begins to miss attacks.

All tests are repeated with background traffic levels of 25%, 50%, 75% and 100% of the maximum throughput of the device, and the total number of exploits detected and blocked is noted. For each type of background traffic, we also determine the maximum load the sensor can sustain before it begins to drop packets/miss alerts.

Any device which permits malicious traffic to pass through the DUT will fail the overall test immediately.

6.1 RAW PACKET PROCESSING PERFORMANCE (UDP TRAFFIC)

This test uses UDP packets of varying sizes generated by Spirent SmartBits traffic generation tools.

A constant stream of the appropriate packet size - with variable source IP addresses and ports transmitting to a single fixed IP address/port - is transmitted bi-directionally through each port pair of the DUT.

Each packet contains dummy data, and is targeted at a valid port on a valid IP address on the target subnet. The percentage load and packets per second (pps) figures across each in-line port pair are verified by the Adtech network monitoring tool before each test begins. Multiple tests are run and averages taken where necessary. Each test is repeated with traffic loads of 25%, 50%, 75% and 100% of the maximum throughput of the DUT, and the percentage of attacks detected and blocked is recorded at each load level. Maximum throughput with zero packet loss is also recorded.

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).

The aim of this test is purely to determine the raw packet processing capability of each in-line port pair of the DUT, and its effectiveness at passing “useless” packets quickly in order to pass potential attack packets to the detection engine.

6.1.1 128 BYTE PACKETS

Maximum 842,000 Packets Per Second per Gigabit of traffic. Repeated with traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT.

6.1.2 256 BYTE PACKETS

Maximum 452,000 Packets Per Second per Gigabit of traffic. Repeated with traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT.

6.1.3 512 BYTE PACKETS

Maximum 235,000 Packets Per Second per Gigabit of traffic. Repeated with traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. This test provides a reasonable indication of the ability of a device to process packets from the wire on an “average” network.

6.1.4 1024 BYTE PACKETS

Maximum 120,000 Packets Per Second per Gigabit of traffic. Repeated with traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT.

6.1.5 1514 BYTE PACKETS

Maximum 82,000 Packets Per Second per Gigabit of traffic. Repeated with traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. This test has been included mainly to demonstrate how easy it is to achieve good results using large packets – beware of test results that only quote performance figures using similar packet sizes.

6.2 MAXIMUM CAPACITY

The use of multiple BreakingPoint appliances allows us to create true “real world” traffic at multi-Gigabit speeds as a background load for our tests.

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 DUT is causing unacceptable increase in open connections on the server-side
- **Excessive response time for HTTP transactions/SMTP sessions** - latency within the DUT is causing excessive delays and increased response time to 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 DUT is causing connections to time out

6.2.1 THEORETICAL MAXIMUM CONCURRENT TCP CONNECTIONS

This test is designed to determine the maximum concurrent TCP connections of the DUT with no data passing across the connections. This type of traffic would not typically be found on a normal network, but it provides the means to determine the maximum possible concurrent connections figure.

A maximum of 7.5 million Layer 4 TCP sessions are opened across the DUT. Each session is opened normally and then held open for the duration of the test as additional sessions are added up to the maximum possible. Load is increased until no more connections can be established, and this number is recorded.

6.2.2 THEORETICAL MAXIMUM CONCURRENT TCP CONNECTIONS WITH DATA

This test is identical to 6.2.1 except that once the maximum number of concurrent connections have been established, 1GB of data is transmitted across them in 21KB segments. This ensures that the DUT is capable of passing data across the connections once they have been established.

6.2.3 MAXIMUM CONCURRENT STATEFUL TCP CONNECTIONS

This test is identical to 6.2.1, but is designed to verify the maximum concurrent TCP connections on which the vendor claims the DUT can maintain state.

A legitimate HTTP session is opened and the first packet of a two-packet exploit is transmitted. As the number of open connections approaches the maximum claimed by the vendor, the initial HTTP session is then completed with the second half of the exploit and the session is closed. If the sensor is still maintaining state on the original session, the exploit will be recorded. If the state tables have been exhausted, the exploit string will be seen as a non-stateful attack, and will thus be ignored.

Both halves of the exploit are required to trigger an alert - a product will fail the test if it fails to generate an alert after the second packet is transmitted, or if it raises an alert on either half of the exploit on its own.

6.2.4 MAXIMUM TCP CONNECTIONS PER SECOND

This test is designed to determine the maximum TCP connection rate of the DUT with one byte of data passing across the connections. This type of traffic would not typically be found on a normal network, but it provides the means to determine the maximum possible TCP connection rate.

A maximum of 750,000 connections per second are generated across the DUT, ramped slowly to determine the exact point of failure. Each session is opened normally, one byte of data passed through the DUT, and then the session is closed immediately. Load is increased until one or more of the breaking points defined earlier is reached.

6.2.5 MAXIMUM HTTP CONNECTIONS PER SECOND

This test is designed to determine the maximum TCP connection rate of the DUT with a 1 byte HTTP response size. The response size defines the number of bytes contained in the body, excluding any bytes

associated with the HTTP header. A 1 byte response size is designed to provide a theoretical maximum HTTP connections per second rate.

Client and server are using HTTP 1.0 without keep alive, and the client will open a TCP connection, send one HTTP request, and close the connection. This ensures that all TCP connections are closed immediately the request is satisfied, thus any concurrent TCP connections will be caused purely as a result of latency within the DUT. Load is increased until one or more of the breaking points defined earlier is reached.

6.2.6 MAXIMUM HTTP TRANSACTIONS PER SECOND

This test is designed to determine the maximum HTTP transaction rate of the DUT with a 1 byte HTTP response size. The object size defines the number of bytes contained in the body, excluding any bytes associated with the HTTP header. A 1 byte response size is designed to provide a theoretical maximum connections per second rate.

Client and server are using HTTP 1.1 with persistence, and the client will open a TCP connection, send ten HTTP requests, and close the connection. This ensures that TCP connections remain open until all ten HTTP transactions are complete, thus eliminating the maximum connection per second rate as a bottleneck (one TCP connection = 10 HTTP transactions). Load is increased until one or more of the breaking points defined earlier is reached.

6.3 BEHAVIOR OF THE STATE ENGINE UNDER LOAD

This test determines whether the DUT is capable of preserving state across a large number of open connections over an extended time period.

At various points throughout the test (including after the maximum has been reached), it is confirmed that the DUT is still capable of detecting and blocking freshly-launched exploits, as well as confirming that the device does not block legitimate traffic (perhaps as a result of state filling up).

6.3.1 ATTACK DETECTION/BLOCKING - NORMAL LOAD

This test determines if the sensor is able to detect and block new exploits as the number of open sessions reaches 75 per cent of the maximum determined in Test 6.2.1.

6.3.2 STATE PRESERVATION - NORMAL LOAD

This test determines if the sensor maintains the state of pre-existing sessions as the number of open sessions reaches 75 per cent of the maximum determined in Test 6.2.1.

A legitimate HTTP session is opened and the first packet of a two-packet exploit is transmitted. As the number of open connections approaches the maximum, the initial HTTP session is then completed with the second half of the exploit and the session is closed. If the sensor is still maintaining state on the original session, the exploit will be recorded. If the state tables have been exhausted, the exploit string will be seen as a non-stateful attack, and will thus be ignored. Both halves of the exploit are required to trigger an alert - a product will fail the test if it fails to generate an alert after the second packet is transmitted, or if it raises an alert on either half of the exploit on its own

6.3.3 PASS LEGITIMATE TRAFFIC - NORMAL LOAD

This test ensures that the sensor continues to pass legitimate traffic as the number of open sessions reaches 75 per cent of the maximum determined in Test 6.2.1.

6.3.4 ATTACK DETECTION/BLOCKING - MAXIMUM EXCEEDED

This test determines if the sensor is able to detect and block new exploits as the number of open sessions exceed the maximum determined in Test 6.2.1.

6.3.5 STATE PRESERVATION - MAXIMUM EXCEEDED

This test determines if the sensor maintains the state of pre-existing sessions as the number of open sessions exceed the maximum determined in Test 6.2.1. Method of execution is identical to Test 6.3.2.

6.3.6 PASS LEGITIMATE TRAFFIC - MAXIMUM EXCEEDED

This test ensures that the sensor continues to pass legitimate traffic as the number of open sessions exceed the maximum determined in Test 6.2.1. **NB:** This is **not** a test fail condition – each vendor must choose whether to block new connections or lose state on existing ones once resources are exhausted. The best solution is to allow the administrator to choose and configure accordingly.

6.4 HTTP CAPACITY WITH NO TRANSACTION DELAYS

The aim of these tests is to stress the HTTP detection engine and 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. This provides a test environment that is as close to “real world” as it is possible to achieve in a lab environment, while ensuring absolute accuracy and repeatability.

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

6.4.1 2,500 CONNECTIONS PER SECOND

Max 2,500 new connections per second per Gigabit of traffic with a 44KB HTTP response size - average packet size 900 bytes - maximum 140,000 packets per second per Gigabit of traffic. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. With relatively low connection rates and large packet sizes, all sensors should be capable of performing well throughout this test.

6.4.2 5,000 CONNECTIONS PER SECOND

Max 5,000 new connections per second per Gigabit of traffic with a 21KB HTTP response size - average packet size 670 bytes - maximum 185,000 packets per second per Gigabit of traffic. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. With average connection rates and average packet sizes, this is a good approximation of a real-world production network, and all sensors should be capable of performing well throughout this test.

6.4.3 10,000 CONNECTIONS PER SECOND

Max 10,000 new connections per second per Gigabit of traffic with a 10KB HTTP response size - average packet size 550 bytes - maximum 225,000 packets per second per Gigabit of traffic. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. With average

packet sizes coupled with very high connection rates this represents a very heavily used production network and is a strenuous test for any sensor.

6.4.4 20,000 CONNECTIONS PER SECOND

Max 20,000 new connections per second per Gigabit of traffic with a 4.5KB HTTP response size - average packet size 420 bytes - maximum 300,000 packets per second per Gigabit of traffic. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. With small packet sizes and extremely high connection rates this is an extreme test for any sensor.

6.4.5 40,000 CONNECTIONS PER SECOND

Max 40,000 new connections per second per Gigabit of traffic with a 1.7KB HTTP response size - average packet size 270 bytes - maximum 445,000 packets per second per Gigabit of traffic. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. With small packet sizes and extremely high connection rates this is an extreme test for any sensor.

6.5 HTTP CAPACITY WITH TRANSACTION DELAYS

This is identical to the previous test except that it includes 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.

6.5.1 5,000 CONNECTIONS PER SECOND WITH DELAY

Max 5,000 new connections per second per Gigabit of traffic with a 21KB HTTP response size - average packet size 670 bytes - maximum 185,000 packets per second per Gigabit of traffic. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. 10 second transaction delay resulting in an additional 50,000 open connections over test 6.4.2. With average connection rates and average packet sizes, this is a good approximation of a real-world production network, and all sensors should be capable of performing well throughout this test.

6.5.2 11KBYTE RESPONSE WITH DELAY

Max 10,000 new connections per second per Gigabit of traffic with a 10KB HTTP response size - average packet size 550 bytes - maximum 225,000 packets per second per Gigabit of traffic. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. 10 second transaction delay resulting in an additional 100,000 open connections over test 6.4.3. With average packet sizes coupled with very high connection rates represents a very heavily used production network and is a strenuous test for any sensor.

6.6 “REAL WORLD” TRAFFIC

Whereas previous tests provide a pure HTTP environment with varying connection rates and average packet sizes, the aim of this test is to simulate a “real world” environment by introducing additional protocols and real content while still maintaining a precisely repeatable and consistent background traffic load (something rarely seen in a real world environment).

The result is a background traffic load that is closer to what may be found on a heavily-utilized “normal” production network.

6.6.1 “REAL WORLD” PROTOCOL MIX (PERIMETER)

Traffic is generated across the DUT comprising the following protocol mix typical of that seen by a perimeter security device:

- *HTTP text* 33%
- *HTTP Images (<50k)* 14%
- *SMTP* 18%
- *FTP* 8%
- *DNS* 6%
- *HTTP Video* 4%
- *HTTP Audio* 4%
- *HTTP Images (>300k)* 4%
- *SSH* 4%
- *AOL IM* 3%
- *SIP/RTP* 1%
- *BitTorrent* 1%

For this test and 6.6.2, HTTP traffic comprises genuine transactions and Web pages from real Web sites such as Google, Yahoo, MSN, NSS Labs, etc. including small (<50KB) and large (>300KB) Jpeg images. Also included as part of the HTTP traffic is genuine QuickTime movie content and MP3 files, taking the total HTTP traffic of all types to approximately 65% of the overall load. SMTP traffic comprises real e-mail messages of varying lengths (with and without attachments) from the NSS Labs mail server.

Maximum 6000 connections per second per Gigabit of traffic - 220,000 packets per second per Gigabit of traffic - average packet size of 550 bytes. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. Maximum of 5,000 open connections during the test.

With lower connection rates, average packets sizes, and a common protocol mix comprising protocols which all require inspection by the NIPS engine, this is a good approximation of a heavily-used production network. All sensors should be capable of performing well throughout this test (and 6.6.2).

6.6.2 “REAL WORLD” PROTOCOL MIX (CORE)

Traffic is generated across the DUT comprising the following protocol mix typical of that seen by a network core security device:

- *HTTP text* 24%
- *SMB File transfer* 14%
- *HTTP Images (<50k)* 12%
- *SMTP* 12%
- *PostgreSQL* 10%
- *DNS* 6%
- *DCERPC* 4%
- *FTP* 3%
- *SMB NULL* 3%

- *HTTP Video* 2%
- *HTTP Audio* 2%
- *HTTP Images (>300k)* 2%
- *AIM* 2%
- *SIP/RTP* 1%
- *NFS* 1%
- *SSH* 1%
- *RTSP* 1%

Maximum 5000 connections per second per Gigabit of traffic - 270,000 packets per second per Gigabit of traffic - average packet size of 440 bytes. Repeated with background traffic loads of 25%, 50%, 75% and 100% of maximum throughput of DUT. Maximum of 6,000 open connections during the test.

6.7 LATENCY

The aim of the latency and user response time tests is to determine the effect the sensor has on the traffic passing through it under various load conditions.

Should a device impose a high degree of latency on the packets passing through it, a network or security administrator would need to think carefully about how many devices could be installed in a single data path before user response times became unacceptable or the combination of devices caused excessive timeouts.

This test uses UDP packets of varying sizes generated by a SmartBits SMB6000 chassis to determine raw packet latency at Layer 2. The Spirent SmartFlow software runs through several iterations of the test, varying the traffic load through multiple in-line port pairs bi-directionally from 25% to 100% of the maximum DUT throughput.

This is repeated for a range of packet sizes (128, 256, 512, 1024 and 1514 bytes) of UDP traffic with variable IP addresses and ports. At each iteration of the test, SmartFlow records the number of packets dropped, together with average and maximum latency, measured in microseconds.

This test - while not indicative of real-life network traffic - provides an indication of how much the sensor affects the traffic flow through it. This data is particularly useful for network administrators who need to gauge the effect of any form of in-line device which is likely to be placed at critical points within the corporate network.

Note that at any packet size NSS considers 1ms to be the acceptable limit for a typical perimeter deployment, and 300µs to be the acceptable limit for the network core.

6.7.1 LATENCY

SmartFlow traffic is passed across the infrastructure switches and through all in-line port pair of the DUT simultaneously (the latency of the basic infrastructure is known and is constant throughout the tests). The packet loss and average latency (µs) are recorded at each packet size (128, 256, 512, 1024 and 1514 bytes) and each load level from 25% to 100% load (in 25% load steps).

6.8 USER RESPONSE TIMES

Multi-protocol sessions are generated through the DUT in order to determine the user experience in terms of failed connections and response times.

6.8.1 APPLICATION RESPONSE TIME

Traffic is generated as per test 6.6.1 (Real World Perimeter Mix) for 15 minutes and the following figures are recorded:

- *TCP Average Time to Open*
- *TCP Average Time to Response Packet*
- *TCP Average Time to Close*
- *Application Average Response Time: HTTP*
- *Application Average Response Time: SMTP*
- *Application Average Response Time: DNS*

7 TEST METHODOLOGY – 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 receive NSS certification.

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

7.1.1 BLOCKING UNDER EXTENDED ATTACK

The DUT is exposed to a constant stream of exploits over an extended period of time. The device is configured to block and alert, and thus this test provides an indication the effectiveness of both the blocking and alert handling mechanisms.

A continuous stream of exploits mixed with legitimate sessions is transmitted through the device at a maximum of 100Mbps (max 50,000 packets per second, average packet sizes in the range of 120-350 bytes) for 8 hours with no additional background traffic. This is not intended as a stress test in terms of traffic load (covered in the previous section) - merely a reliability test in terms of consistency of blocking performance.

The device is expected to remain operational and stable throughout this test, and to block 100 per cent of recognisable exploits, raising an alert for each. If any recognisable exploits are passed - caused by either the volume of traffic or the sensor failing open for any reason - this will result in a FAIL.

7.1.2 PASSING LEGITIMATE TRAFFIC UNDER EXTENDED ATTACK:

This test is identical to 7.1.1, where we expose the external interface of the device to a constant stream of exploits over an extended period of time.

The device is expected to remain operational and stable throughout this test, and to pass most/all of the legitimate traffic. If an excessive amount of legitimate traffic is blocked throughout this test - caused by either the volume of traffic or the sensor failing closed for any reason - this will result in a FAIL.

7.1.3 PROTOCOL FUZZING

This test stresses the protocol stacks of the DUT by exposing it to traffic from various protocol randomizer tools. Several of the tools in this category are based on the ISIC test suite and the BreakingPoint *Stack Scrambler* component.

Traffic load is a maximum of 350Mbps and 60,000 packets per second (average packet size is 690 bytes). Results are presented as a simple PASS/FAIL - the device is expected to remain operational and capable of detecting and blocking exploits throughout the test.

7.1.4 PROTOCOL MUTATION

This test stresses the protocol stacks of the DUT by exposing it to traffic from various protocol mutation tools. Several of the tools in this category are based on the Mu Security Analyzer and the BreakingPoint *Stack Scrambler* component.

7.1.5 POLICY PUSH

This test is identical to 6.8.1. A new policy is pushed to the DUT during the test, and the application response times are recorded for comparison with Test 6.8.1 to determine the effect of pushing policies on legitimate traffic.

7.1.6 POWER FAIL

This test is identical to 6.8.1. Power to the DUT is cut during the test, and the application response times are recorded for comparison with Test 6.8.1 to determine the effect of power failure on legitimate traffic.

If the device is configured to fail open, there should be minimal loss of legitimate sessions throughout the test (over and above the baseline loss expected through switch renegotiation). If the device is configured to fail closed, no traffic should be passed once power has been cut.

7.1.7 REDUNDANCY

Does the DUT include multiple redundant critical components (fans, power supplies, hard drive, etc.) (YES/NO/OPTION).

7.1.8 FAIL OPEN (POWER FAIL/REBOOT)

Does the DUT provide the ability to fail open with minimal/zero loss of legitimate traffic (either via built-in, or optional hardware bypass) during power fail and reboot (YES/NO/OPTION).

7.1.9 FAIL OPEN (RESOURCE ISSUES)

Does the DUT provide the ability to pass all traffic when resources are exhausted or it is no longer possible to analyze traffic for any reason (i.e. packet rate exceeds device capabilities).

7.1.10 FAIL CLOSED (POWER FAIL/REBOOT)

Does the DUT provide the ability to fail closed during power fail and reboot (YES/NO/OPTION).

7.1.11 FAIL CLOSED (RESOURCE ISSUES)

Does the DUT provide the ability to block all traffic when resources are exhausted or it is no longer possible to analyze traffic for any reason (i.e. packet rate exceeds device capabilities).

7.1.12 HIGH AVAILABILITY (HA) OPTION (STATEFUL)

Is an HA option available for this device, providing fully stateful active-active or active-passive failover between devices (YES/NO).

7.1.13 HIGH AVAILABILITY (HA) OPTION (NON-STATEFUL)

Is an HA option available for this device, providing any form of failover between devices where existing connections may be lost during failover (YES/NO).

7.1.14 PERSISTENCE OF DATA

The DUT should retain all configuration data, policy data and locally logged data once restored to operation following power failure.

7.1.15 IPV6

The DUT should be capable of detecting exploits over both IPV6 and IPV4.

8 TEST METHODOLOGY – MANAGEMENT & CONFIGURATION

This section evaluates the features and usability of the DUT and associated management infrastructure.

8.1 MANAGEMENT PORT

Clearly the ability to manage the alert data collected by the sensor is a critical part of any IDS/NIPS system. For this reason, an attacker could decide that it is more effective to attack the management interface of the device than the detection interface.

Given access to the management network, this interface is often more visible and more easily subverted than the detection interface, and with the management interface disabled, the administrator has no means of knowing his network is under attack.

8.1.1 OPEN PORTS REQUIRED

The vendor will list the open ports and active services on the management interface along with their use.

8.1.2 OPEN PORTS DETECTED

The management port will be scanned to determine ports/services visible on the management interface. If any ports additional to those listed in Test 8.1.1 are discovered, this will result in an automatic FAIL.

8.1.3 PROTOCOL FUZZING

This test stresses the protocol stacks of the DUT management interface by exposing it to traffic from various protocol randomizer tools. Several of the tools in this category are based on the ISIC test suite and the BreakingPoint *Stack Scrambler* component.

Traffic load is a maximum of 350Mbps and 60,000 packets per second (average packet size is 690 bytes). Results are presented as a simple PASS/FAIL - the device is expected to remain (a) operational and capable of detecting and blocking exploits, and (b) capable of communicating in both directions with the management server/console throughout the test.

8.1.4 PROTOCOL FUZZING DETECTION ON MANAGEMENT PORT

Are fuzzing attempts attacks detected by the DUT even though targeted at the management port (YES/NO).

8.2 MANAGEMENT & CONFIGURATION - GENERAL

In addition to the specific tests noted below, NSS will execute an in-depth technical evaluation covering all the main features and benefits of the NIPS product under test. The accompanying report will fully evaluate each product in terms of ease of use, management and configuration, and alerting and reporting capabilities.

8.2.1 TRANSPARENT MODE

The DUT must be capable of running in transparent bridge mode (Layer 2), with no IP address assigned to detection ports. Detection ports should ignore all direct connection attempts.

8.2.2 MANAGEMENT PORT

The DUT should 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 cause for failure providing management connection and communication is securely encrypted.

8.2.3 MANAGEMENT PROTOCOL

Connection from management console to DUT should be protected by a minimum of a user name/password combination or multi-factor authentication system, and all communications between should be 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.

8.2.4 AUTHENTICATION

Access to management console should be 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.

8.2.5 ENTERPRISE AUTHENTICATION

Access to management console should be protected by a granular user authentication system which allows for restriction of individual users to specific devices, ports, reports, alerts and security policies. Authenticated users should be unable to access devices/ports/policies/alerts/reports/etc. restricted to other users of the system.

8.2.6 DIRECT DUT MANAGEMENT (OPTIONAL)

Direct access to the DUT should be provided (either via command line or Web interface) for single-device management.

8.2.7 CENTRALIZED DUT MANAGEMENT

A centralized management system should be provided to manage one or more NIPS 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.

8.2.8 PASS-THROUGH MODE (OPTIONAL)

It should be 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 NIPS sensor 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).

8.2.9 SIGNATURE UPDATE

The vendor should demonstrate access to a vulnerability research capability (either in-house or via a recognized third-party) which is able to provide timely and accurate signature updates at regular intervals.

8.2.10 SECURE DUT REGISTRATION

Initial registration of DUT to central management console should be in a fully secure manner (it is permitted to offer a less secure/rapid option, but this should not be the default).

8.2.11 DOCUMENTATION

Adequate documentation should be provided for both installation, and day-to-day management.

8.3 MANAGEMENT & CONFIGURATION – POLICY

8.3.1 DUT CONFIGURATION

The management system should provide the means to configure one or more sensors from a central location, assigning signatures, sensor settings, etc.

8.3.2 POLICY DEFINITION

The management system should provide the means to define and save multiple security policies, consisting of:

- *General sensor configuration*
- *System-wide parameters*
- *Signatures enabled/disabled*
- *Actions to take when malicious traffic discovered*

8.3.3 RECOMMENDED SETTINGS

The vendor should provide a default policy or suite of recommended 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.).

8.3.4 CUSTOM ATTACK SIGNATURES

It should be possible for the administrator to be able to define custom attack signatures for use in standard policies.

8.3.5 BULK OPERATIONS

It should be 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.).

8.3.6 GRANULARITY

The DUT should be capable of blocking or creating exceptions based on IP address, application, 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.).

8.3.7 POLICY ASSOCIATION

Once policies have been defined, it should be possible to associate them with specific DUT or groups of DUTs.

8.3.8 INHERITANCE

It should be 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.

8.3.9 VIRTUALIZATION

Once policies have been defined, it should be 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.

8.3.10 POLICY DEPLOYMENT

Once policies have been defined, it should be possible to distribute them to the appropriate device(s), virtual device(s), or groups of devices in a single operation.

8.3.11 POLICY AUDITING

All changes to policies should be 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 (OPTIONAL) the system should record the actual changes.

8.3.12 POLICY VERSION CONTROL

All changes to policies should be recorded by saving a version of the policy before each change. It should be possible to roll-back to a previous version of any policy via a single operation.

8.4 MANAGEMENT & CONFIGURATION - ALERT HANDLING

8.4.1 REQUIRED LOG EVENTS

The DUT should 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*

8.4.2 LOG LOCATION (OPTIONAL)

The log events should be logged on the DUT initially, in a secure manner, and subsequently transmitted to a central repository for permanent storage.

8.4.3 COMMUNICATION INTERRUPTION

Where communications between sensor and console/management server are interrupted, storage capacity on the DUT should be sufficient to hold one week’s worth of log data on a typical network. 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. This option should be configurable by the administrator.

8.4.4 LOG FLOODING

Mechanisms should be 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. (It should be possible to disable aggregation/flood protection completely for testing purposes to ensure NSS engineers can see every individual alert).

8.4.5 ALERTS

The DUT should record log entries each time it detects malicious traffic. At a minimum (depending on protocol), these log entries should contain:

- *Unique event ID*
- *Date and time of event*
- *DUT ID (includes sensor ID, port ID, etc.)*
- *Direction of traffic (physical/logical source and destination interfaces)*
- *Detection engine which raised the alert (OPTIONAL)*
- *Source IP address*
- *Source port/service (where applicable)*
- *Destination IP address*
- *Destination port/service (where applicable)*
- *ICMP message type and code (where applicable)*
- *Protocol*
- *Unique signature ID*
- *Human-readable description of the event/exploit*
- *CVE reference, Bugtraq ID, or other non-vendor-specific identifier*
- *Action taken by the DUT (block, log, etc.)*

8.4.6 ALERT ACCURACY

The DUT should record log entries which 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.

8.4.7 CENTRALIZED ALERTS

Regardless of how many sensors are installed, all alerts should be delivered to, and handled by, a single, central, management console. From that console, it should be possible to view all alerts globally, or select alerts from individual devices (logical or physical).

8.4.8 ALERT DELIVERY MECHANISM

At a minimum, the DUT should be able to 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.

8.4.9 ALERT ACTIONS (MANDATORY)

On detecting malicious traffic, the DUT should be able to perform the following actions at a minimum:

- *Ignore*

- *Log only*
- *Drop packet (no reset)*
- *Drop session (no reset)*
- *E-mail administrator*

8.4.10 ALERT ACTIONS (OPTIONAL)

On detecting malicious traffic, the DUT may optionally be able to perform the following actions:

- *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 external firewall*
- *Reconfigure switch to isolate/quarantine offending port*
- *Page administrator*

8.4.11 FORENSIC ANALYSIS

The DUT should provide the ability to capture individual packets, a range of packets, or an entire session where required (globally, or on a rule-by-rule basis).

8.4.12 SUMMARIZE ALERTS

The central console should 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.

8.4.13 VIEW ALERT DETAIL

The central console should provide the ability to select an individual alert and view the following information at a minimum:

- *Detailed alert data (including all data mentioned in Test 8.4.5)*
- *Detailed exploit data (description of the exploit research)*
- *Signature/rule*
- *Remediation data/preventative action*

8.4.14 VIEW RELATED POLICY

Having selected an alert, the system should provide the ability to access directly the policy and rule which triggered the event in order to view and/or modify the policy for further fine tuning.

8.4.15 VIEW PACKET CONTENTS

The central console should provide the ability to select an individual alert and view the contents of the trigger packet or context data for the exploit.

8.4.16 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.

8.4.17 CORRELATION (AUTOMATIC)

The system should provide the means to infer connections between multiple alerts and group them together as incidents automatically.

8.4.18 CORRELATION (MANUAL)

The system should provide the means for the administrator to infer connections between multiple alerts and group them together as incidents manually.

8.4.19 INCIDENT WORKFLOW

The system should provide the ability to annotate and track incidents to resolution.

8.5 MANAGEMENT & CONFIGURATION – REPORTING

8.5.1 CENTRALIZED REPORTS

No matter how many sensors are installed, the system should be capable of reporting on all alerts from a single, central, management console. From that console, it should be possible to report all alerts globally, or to report on alerts from individual devices (logical or physical).

8.5.2 TOP ATTACKS

The system should provide a report listing the top N attacks in the previous hour, day, week, month, year, or custom date range.

8.5.3 TOP SOURCES

The system should provide a report listing the top N source IPs from which attacks have been detected in the previous hour, day, week, month, year, or custom date range.

8.5.4 TOP TARGETS

The system should provide a report listing the top N target IPs at which attacks have been launched in the previous hour, day, week, month, year, or custom date range.

8.5.5 TOP SERVICES

The system should provide a report listing the top N target ports/services at which attacks have been launched in the previous hour, day, week, month, year, or custom date range.

8.5.6 TOP PROTOCOLS

The system should provide a report listing the top N protocols over which attacks have been launched in the previous hour, day, week, month, year, or custom date range.

8.5.7 CUSTOM REPORTS

The report generator should provide the ability to construct complex data filters in a search form and summarize alerts on the specified search criteria.

8.5.8 SAVED REPORTS

Having defined a custom report filter, it should be possible to save it for subsequent recall.

8.5.9 SCHEDULED REPORTS

It should be possible to schedule saved reports for regular unattended runs. The output should be saved as HTML or PDF at a minimum. It should optionally be possible to publish to a central FTP/Web server, and/or e-mail reports to specified recipients.

8.5.10 LOG FILE MAINTENANCE

The system should provide for automatic rotation of log files, archiving, restoring from archive, and reporting from archived logs.

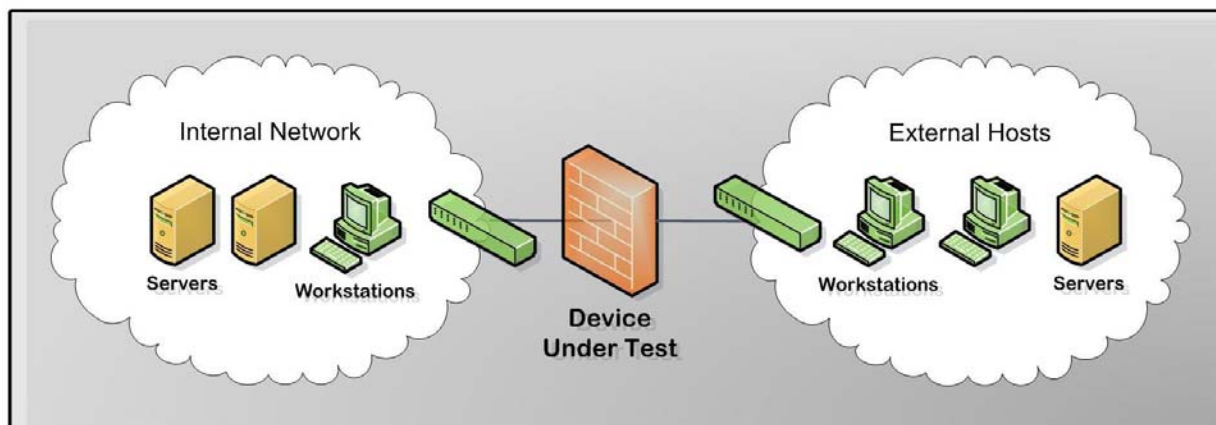
APPENDIX A: NETWORK IPS TEST ENVIRONMENT

The aim of this procedure is to provide a thorough test of all the main components of an in-line NIPS device in a controlled and repeatable manner and in the most “real world” environment that can be simulated in a test lab.

The Test Environment

The NSS Labs test network is a multi-Gigabit infrastructure based around multiple Cisco Catalyst 6500-series switches (these have a mix of fiber and copper Gigabit interfaces). The NIPS will be configured for the use-case appropriate to the target deployment environment.

Traffic generation equipment - such as the hosts generating exploits, Spirent Avalanche, TestCenter and Smartbits transmit ports - is connected to the “external” network, while the “receiving” equipment - such as the vulnerable hosts for the exploits, Spirent Reflector, TestCenter Spirent Smartbits receive ports - is connected to the internal network. The NIPS is connected between two “gateway” switches - one at the edge of the external network, and one at the edge of the external network.



All “normal” network traffic, background load traffic and exploit traffic will therefore be transmitted through the NIPS, from external to internal (responses will flow in the opposite direction). The same traffic is mirrored to multiple SPAN ports of the external gateway switch, to which Adtech AX/4000 network monitoring devices are connected. The Adtech AX/4000’s monitor the same mirrored traffic to ensure that the total amount of traffic per in-line port pair never exceeds 1Gbps.

The management interface is used to connect the appliance to the management console on a private subnet. This ensures that the sensor and console can communicate even when the target subnet is subjected to heavy loads, in addition to preventing attacks on the console itself.

APPENDIX B: TEST INFRASTRUCTURE

Special thanks go to our test infrastructure partners who provide much of the equipment, software, and support that make this testing possible:

