



NETWORK INTRUSION PREVENTION SYSTEMS

TEST METHODOLOGY V6.1

To receive a licensed copy or report misuse,
Please contact NSS Labs at: +1 512-961-5300
or advisor@nsslabs.com

© 2010 NSS Labs, Inc. 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.

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 Methodology is conditioned on the following:

1. The information in this Methodology is subject to change by NSS Labs without notice.
2. The information in this Methodology is believed by NSS Labs to be accurate and reliable, but is not guaranteed. All use of and reliance on this Methodology 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 Methodology 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 Methodology 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.

CONTACT INFORMATION

NSS Labs

P.O. Box 130573
Carlsbad, CA 92013 USA
+1 (512) 961-5300
info@nsslabs.com
www.nsslabs.com

TABLE OF CONTENTS

1	<i>Introduction</i>	5
1.1	About NSS Labs	5
1.2	The Need for Intrusion Prevention	5
1.3	The Need for Testing	5
1.4	About This Test Methodology and Report	5
1.5	Tested Products	6
1.6	Non-Participating Vendors	7
1.7	About Tuning and Configuration	7
2	<i>Product Comparisons</i>	8
2.1	Security Effectiveness	8
2.1.1	Protection with Default Policy Settings.....	8
2.1.2	Protection with Tuned Policy.....	8
2.1.3	Protection: Default vs. Tuned Policies.....	8
2.2	Effectiveness by Attack Vector	8
2.3	Effectiveness by Disclosure Date	8
2.3.1	Protection by Year – Tuned Policy.....	8
2.3.2	Protection by Year – Default vs. Tuned Policies.....	8
2.4	Resistance to Evasion	8
2.4.1	Impact of Evasion – Default Policy.....	9
2.5	Performance	9
2.6	Total Cost of Ownership	9
2.6.1	Labor per Product (in Hours).....	9
2.6.2	Purchase Price and Total Cost of Ownership.....	10
2.6.3	Value: Cost per Mbps and Exploit Blocked – Tuned Policy.....	10
2.6.4	Three-Year Total Cost of Ownership per Protected Mbps.....	10
2.6.5	Comparison by Performance Category.....	10
3	<i>Product Guidance</i>	11
3.1	Recommend	11
3.2	Neutral	11
3.3	Caution	11
4	<i>Methodology Elements Overview</i>	12
5	<i>Security Effectiveness</i>	19
5.1	Detection Engine	19
5.1.1	System Exposure.....	19
5.1.2	Service Exposure.....	19
5.1.3	System or Service Fault.....	19
5.2	Threat Vector	20
5.2.1	Attacker Initiated.....	20

5.2.2	Target Initiated	20
5.2.3	Network	20
5.2.4	Local	20
5.3	Target Type.....	20
5.4	Coverage by Result	20
5.4.1	Arbitrary Code Execution	21
5.4.2	Buffer Overflow	21
5.4.3	Code Injection.....	21
5.4.4	Cross-Site Script	21
5.4.5	Directory Traversal	21
5.4.6	Privilege Escalation	21
5.5	Coverage by Vendor.....	21
5.6	Evasion	22
5.6.1	Unmodified Exploit Validation	22
5.7	Packet Fragmentation.....	22
5.8	Stream Segmentation	23
5.9	RPC Fragmentation	23
5.10	URL Obfuscation	24
5.11	HTML Obfuscation.....	24
5.12	FTP Evasion.....	26
6	Performance	27
6.1	Raw Packet Processing Performance (UDP Traffic).....	27
6.1.1	128 Byte Packets.....	28
6.1.2	256 Byte Packets.....	28
6.1.3	512 Byte Packets.....	28
6.1.4	1024 Byte Packets	28
6.1.5	1514 Byte Packets	28
6.2	Connection Dynamics – Concurrency and Connection Rates	28
6.2.1	Theoretical Maximum Concurrent TCP Connections	29
6.2.2	Theoretical Maximum Concurrent TCP Connections with Data.....	29
6.2.3	Maximum Concurrent Stateful TCP Connections.....	29
6.2.4	Maximum TCP Connections per Second	29
6.2.5	Maximum HTTP Connections per Second	29
6.2.6	Maximum HTTP Transactions per Second.....	30
6.3	Behavior of the State Engine Under Load	30
6.3.1	Attack Detection/Blocking - Normal Load	30
6.3.2	State Preservation - Normal Load	30
6.3.3	Pass Legitimate Traffic - Normal Load	30
6.3.4	Attack Detection/Blocking - Maximum Exceeded.....	31
6.3.5	State Preservation - Maximum Exceeded	31
6.3.6	Pass Legitimate Traffic - Maximum Exceeded	31
6.4	HTTP Capacity with No Transaction Delays	31
6.4.1	44 Kbyte HTTP Response.....	31

6.4.2	21 Kbyte HTTP Response	31
6.4.3	10 Kbyte HTTP Response	31
6.4.4	4.5 Kbyte HTTP Response	32
6.4.5	1.7 Kbyte HTTP Response	32
6.5	HTTP Capacity with Transaction Delays	32
6.5.1	21 Kbyte HTTP Response with Delay	32
6.5.2	10 Kbyte HTTP Response with Delay	32
6.6	"Real-World" Traffic	32
6.6.1	"Real-World" Protocol Mix (Perimeter)	33
6.6.2	"Real-World" Protocol Mix (Core)	33
7	Management and Configuration Costs	35
7.1	Ease-of-Use	35
7.1.1	Initial Setup (Hours)	35
7.1.2	Time Required for Upkeep (Hours per Year)	35
7.1.3	Time Required to Tune (Hours per Year)	35
7.2	Expected Costs	35
7.2.1	Initial Purchase	35
7.2.2	Ongoing Maintenance & Support (Annual)	35
7.2.3	Installation Labor Cost (@\$75/hr)	35
7.2.4	Management Labor Cost (per Year @\$75/hr)	35
7.2.5	Tuning Labor Cost (per Year @\$75/hr)	35
7.3	Total Cost of Ownership	35
7.3.1	Year 1	35
7.3.2	Year 2	35
7.3.3	Year 3	35
7.3.4	Three-Year Total Cost of Ownership	35
Appendix A: Network IPS Test Environment		36
Appendix B: Special Thanks		37

1 INTRODUCTION

1.1 ABOUT NSS LABS

NSS Labs performs expert, independent security product evaluations and certifications to assist IT teams in selecting and managing the right security products for their environment. We operate the largest security and performance lab in the world. Our test reports and certifications are highly regarded by information security professionals for their rigor, depth, and integrity, and are used to validate purchasing decisions in global organizations.

1.2 THE NEED FOR INTRUSION PREVENTION

As cyber-criminals have become more aggressive over the past years, they have increasingly targeted corporate assets: servers, web-enabled applications and databases, clients, browsers, and plug-ins. Security researchers all agree, the problem is getting worse. The rising number of vulnerability disclosures in widely-deployed operating systems, applications, and appliances is a multi-faceted problem.

Network IPS are an important part of the solution. Designed to identify and block attacks against internal computing assets, a good IPS can provide temporary protection and relief from the immediate need to patch affected systems. The IPS must catch sophisticated attacks while producing nearly zero false positives. And it must not degrade network performance or it will never be installed.

1.3 THE NEED FOR TESTING

Over the past few years of working with clients and testing IPS products, we began to notice some troubling trends. Some vendors missed more attacks than seemed acceptable. A number of vendors refused to participate in our testing—even at no cost. Enterprise readers asked some tough questions about current products and issues. And breaches resulting in compromised data continued to increase.

Meanwhile, we heard several opinions from our clients, industry analysts, and researchers. These beliefs can be summarized as follows:

- IPS is a mature market. There is relatively little difference between products; thus, management and price are the key purchasing factors
- Best-of-breed products are more effective than those from strategic vendors who provide a wider range of products
- The market leader (from an installed base perspective) provides the best protection
- Organizations are protected as long as they keep their IPS systems updated

This test methodology sets out to determine if these beliefs are correct.

1.4 ABOUT THIS TEST METHODOLOGY AND REPORT

NSS Labs' test reports are designed to address the challenges faced by IT professionals in selecting and managing security products. The scope of this particular report is necessarily more limited than our standard certifications and is focused on:

- Security effectiveness
- Performance
- TCO (new in this report)

Live Exploit Testing: NSS Labs' security effectiveness testing leverages deep expertise of our engineers utilizing multiple commercial, open source and proprietary tools as appropriate. With over 1,159 live exploits. This is the industry's most comprehensive test to date. Most notable, all of the live exploits and payloads in our test have been validated in our lab such that:

- a reverse shell is returned
- a bind shell is opened on the target allowing the attacker to execute arbitrary commands
- a malicious payload is installed
- a system is rendered unresponsive
- etc.

This test goes far beyond replaying PCAPs or pressing the button on a test tool. In short, our engineers triggered vulnerabilities for the purpose of validating that an exploit was able to pass through the device under test.

In-line IPS devices exhibit an inverse correlation between security effectiveness and performance. The more deep packet inspection is performed, the fewer packets can be forwarded. Furthermore, it is important to consider a real-world mix of traffic that a device will encounter. NSS Labs utilizes a range of traffic types and mixes.

Since considerable differences exist between default and tuned settings, it is important to measure the associated costs of achieving an elevated security score. Our team timed the vendor experts configuring and tuning their products and applied industry standard cost models to estimate associated costs. Contact us to receive a copy of a helpful spreadsheet you may use to customize these settings for your environment.

This group test methodology is based on years of testing experience and summarizes thousands of individual test cases. Given the overwhelming breadth and depth of this methodology, not all data could be represented comparatively. Please reference the following additional documents for more information:

Additional Reports and Documents:

- Individual Product Reports contain detailed results from the extensive testing by product configuration
- Exposure Reports provide a searchable database of exposed vulnerabilities not protected by IPS products (listed by CVE) to help organizations identify and mitigate specific risks to assets
- Product Guidance Ratings at www.nsslabs.com/resources/product-guidance-ratings.html provide more information about our Recommend, Neutral and Caution ratings.

1.5 TESTED PRODUCTS

In order to garner the greatest participation, and allay any potential concerns of bias, we invited all leading vendors to submit products at no cost. NSS Labs does not track market share or non-technical aspects of vendor products. As input to our selection criteria, we considered vendors we had previously tested, along with the rankings of other traditional analyst firms.

1.6 NON-PARTICIPATING VENDORS

Several vendors (named in report) declined multiple invitations to submit their IPS products at no cost. Organizations using these products are doing so without the benefit of the industry's most rigorous testing. Effectiveness claims should be viewed with a fair amount of skepticism until these products can be further evaluated.

1.7 ABOUT TUNING AND CONFIGURATION

Vendors were encouraged to tune their IPS if they felt their default pre-defined settings were not optimal. Where custom tuning was applied, NSS Labs will make those settings available for readers upon request. Some vendors provide customers with factory-tuned policies instead of manually tuning their products. These policies usually contain all the signatures that can be safely deployed without triggering false positives. This approach allows customers to quickly deploy a more aggressive security policy with relatively little effort or risk and will be reflected positively in the time and cost sections of this test. Once testing begins, the product version will be frozen to preserve the integrity of the test.

2 PRODUCT COMPARISONS

2.1 SECURITY EFFECTIVENESS

Security products are growing increasingly complex and vendors are responding by simplifying the user interface and security policy selection to meet the usability needs of a broadening user base. Indeed, many organizations accept and deploy the default settings, understanding these to be the best recommendations from the vendor.

2.1.1 PROTECTION WITH DEFAULT POLICY SETTINGS

2.1.2 PROTECTION WITH TUNED POLICY

2.1.3 PROTECTION: DEFAULT VS. TUNED POLICIES

2.2 EFFECTIVENESS BY ATTACK VECTOR

Exploits can be initiated either locally by the target (client) or remotely by the attacker. Since 2007, we have seen a dramatic rise in the number of client-side exploits, as these can be easily launched by an unsuspecting user who visits an infected website. IPS products have traditionally not focused on these types of attacks, and the industry has been catching up in response.

NSS utilizes the following definitions:

- **Attacker Initiated:** The threat/exploit is executed by the attacker remotely against a vulnerable application and/or operating system.
- **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.

2.3 EFFECTIVENESS BY DISCLOSURE DATE

The old attacks are still relevant and must be protected against. Different vendors take different approaches to adding coverage once a vulnerability is disclosed. The result is varying levels of protection for vulnerabilities.

2.3.1 PROTECTION BY YEAR – TUNED POLICY

2.3.2 PROTECTION BY YEAR – DEFAULT VS. TUNED POLICIES

2.4 RESISTANCE TO EVASION

Evasion techniques are means of disguising and modifying attacks in order to avoid detection and blocking by security products. Missing a type of evasion means a hacker can use an entire class of exploits to circumvent the IPS, rendering it virtually useless. The techniques used in this test have been widely known for years and should be considered minimum requirements for the IPS product category.

Providing exploit protection results without factoring evasion in can be misleading since the more types of evasion that are missed—IP Fragmentation, TCP Segmentation, RPC Fragmentation, URL Obfuscation, and FTP Evasion—the worse the situation. It is better to miss five evasion techniques in one category (say FTP evasion) than one in each category. Furthermore, missing either IP

Fragmentation or TCP Segmentation is worse than missing any of the other three types of obfuscation since they impact ALL exploits.

2.4.1 IMPACT OF EVASION – DEFAULT POLICY

IP Fragmentation and TCP Segmentation evasions are by far the worst since, if an attacker can avoid detection by fragmenting IP Packets or segmenting TCP Streams, an IPS will be completely blind to ALL attacks. In addition, ready-made tools are available to help attackers with these evasion techniques. Nearly as bad are RPC Fragmentation, URL Obfuscation and FTP / Telnet Evasion since those represent some of the most popular applications on a network.

Thus, missing a single evasion technique opens wider holes for attackers to get through and vendors should rectify such omissions immediately. Any financially-motivated hacker with basic skills will know how to take advantage of these weaknesses, and simple toolkits exist to assist them. Further analysis of evasion techniques is provided in the Exposure Report, including more advanced evasions such as JavaScript evasions and PDF Evasions.

2.5 PERFORMANCE

NSS Labs collected extensive performance metrics during this test, according to our established methodology. The volumes of data produced by these tests are designed to capture maximum capacities or “the edge of performance” that may be obtainable for a given metric. In addition, our real-world traffic mix testing methods enable us to more accurately estimate the performance users can expect in their environments. Due to space considerations and the number of different products, we have summarized some of the most important figures that a network administrator should consider when sizing a deployment.

In general, having more active signatures on a product can have an unfavorable impact on performance. This is a necessary trade-off. The following chart illustrates the range of throughput an administrator can expect to sustain using either the default or tuned policies.

Beyond overall throughput of the device, connection dynamics can play an important role in sizing a security device that will not unduly impede the performance of a system or an application. Below is a subset of figures from our performance tests.

2.6 Total Cost of Ownership

IPS implementations can be complex projects with several factors affecting the overall cost of deployment, maintenance and upkeep. All of these should be considered over the course of the useful life of the solution.

- Product Purchase – the cost of acquisition
- Product Maintenance – the fees paid to the vendor
- Installation – the time required to take the device out of the box, configure it, put it into the network, apply updates and patches, initial tuning, and set up desired logging and reporting.
- Upkeep – the time required to apply periodic updates and patches from vendors, including hardware, software, and protection (signature/filter/rules) updates.
- Tuning – the time required to configure the policy such that the best possible protection is applied while reducing or eliminating false alarms and false positives.

2.6.1 LABOR PER PRODUCT (IN HOURS)

2.6.2 PURCHASE PRICE AND TOTAL COST OF OWNERSHIP

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

2.6.3 VALUE: COST PER MBPS AND EXPLOIT BLOCKED – TUNED POLICY

2.6.4 THREE-YEAR TOTAL COST OF OWNERSHIP PER PROTECTED MBPS

- *Price/Mbps-Protected = Three-Year TCO/(Protection x Throughput).*

This formula discounts the throughput based upon the effectiveness of the device at blocking attacks in order to provide a weighted value.

2.6.5 COMPARISON BY PERFORMANCE CATEGORY

3 PRODUCT GUIDANCE

NSS Labs issues summary product guidance based on evaluation criteria that is important to information security professionals. The evaluation criteria are weighted as follows:

1. **Security effectiveness** - The primary reason for buying an IPS is to achieve a high percentage coverage of common threats
2. **Resistance to evasion** - Failure in any evasion class permits attackers to circumvent protection
3. **Simplicity of management** - In particular, how difficult is it to configure the highest degree of protection
4. **Value** – Customers should seek low TCO and high effectiveness and performance rankings

Products are listed in rank order according to their guidance rating.

3.1 RECOMMEND

A Recommend rating from NSS Labs indicates that a product has performed well and deserves strong consideration. Only the top technical products earn a Recommend rating from NSS Labs—regardless of market share, company size, or brand recognition.

3.2 NEUTRAL

A Neutral rating from NSS Labs indicates that a product has performed reasonably well and should continue to be used if it is the incumbent within an organization. Products that earn a Neutral rating from NSS Labs deserve consideration during the purchasing process.

3.3 CAUTION

A Caution rating from NSS Labs indicates that a product has performed poorly. Organizations using one of these products should review their security posture and other threat mitigation factors, including possible alternative configurations and replacement. Products that earn a Caution rating from NSS Labs should not be short-listed or renewed.

4 METHODOLOGY ELEMENTS OVERVIEW

The following table lists the individual tests NSS Labs performed on each of the products. Direct references are provided to NSS Labs Test IDs from Sections 5 through 7 of our full certification methodology. The detailed results per product are available separately.

Test ID	Description	Result
5.1	Detection Engine	
5.1.1	System Exposure	
5.1.2	Service Exposure	
5.1.3	System or Service Fault	
5.2	Threat Vectors	
5.2.1	Attacker Initiated	
5.2.2	Target Initiated	
5.2.3	Network	
5.2.4	Local	
5.3	Target Type	
5.3.1	Web Server	
5.3.2	Web Browser	
5.3.3	ActiveX	
5.3.4	JavaScript	
5.3.5	Browser Plug-ins / Add-ons	
5.4	Coverage by Result	
5.4.1	Arbitrary Code Execution	
5.4.2	Buffer Overflow	
5.4.3	Code Injection	
5.4.4	Cross-Site Script	
5.4.5	Directory Traversal	
5.4.6	Privilege Escalation	
5.5	Coverage by Vendor	
5.5.1	3Com	
5.5.2	Adobe	
5.5.3	Alt-N	
5.5.4	Apache	
5.5.5	Apple	
5.5.6	Atrium	
5.5.7	Avast	
5.5.8	BEA	
5.5.9	BitDefender	
5.5.10	Borland	

Test ID	Description	Result
5.5.11	CA	
5.5.12	Cisco	
5.5.13	Citrix	
5.5.14	ClamAV	
5.5.15	EMC	
5.5.16	Facebook	
5.5.17	GNU	
5.5.18	Google	
5.5.19	HP	
5.5.20	IBM	
5.5.21	IPSwitch	
5.5.22	ISC	
5.5.23	Kaspersky	
5.5.24	LanDesk	
5.5.25	lighttpd	
5.5.26	Linux	
5.5.27	Macromedia	
5.5.28	MacroVision	
5.5.29	Mailenable	
5.5.30	McAfee	
5.5.31	Mercury	
5.5.32	Microsoft	
5.5.33	MIT	
5.5.34	Mozilla	
5.5.35	Mplayer	
5.5.36	Multiple Vendors	
5.5.37	MySQL	
5.5.38	NOD32	
5.5.39	Novell	
5.5.40	Nullsoft	
5.5.41	OpenLDAP	
5.5.42	OpenOffice	
5.5.43	OpenSSH	
5.5.44	OpenSSL	
5.5.45	Oracle	
5.5.46	Other Misc	
5.5.47	Panda	
5.5.48	RealNetworks	

Test ID	Description	Result
5.5.49	Samba	
5.5.50	SAP	
5.5.51	Snort	
5.5.52	Sophos	
5.5.53	SpamAssassin	
5.5.54	Squid	
5.5.55	Sun Microsystems	
5.5.56	Symantec	
5.5.57	Trend Micro	
5.5.58	Trillian	
5.5.59	UltraVNC	
5.5.60	Veritas	
5.5.61	VideoLan	
5.5.62	VMWare	
5.5.63	WinAmp	
5.5.64	WinFTP	
5.5.65	Winzip	
5.5.66	Yahoo	
5.6	Evasion	
5.6.1	Evasion	
5.7	Packet Fragmentation	
5.7.1	Ordered 8 byte fragments	
5.7.2	Ordered 24 byte fragments	
5.7.3	Out of order 8 byte fragments	
5.7.4	Ordered 8 byte fragments, duplicate last packet	
5.7.5	Out of order 8 byte fragments, duplicate last packet	
5.7.6	Ordered 8 byte fragments, reorder fragments in reverse	
5.7.7	Ordered 16 byte fragments, fragment overlap (favor new)	
5.7.8	Ordered 16 byte fragments, fragment overlap (favor old)	
5.7.9	Out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery	
5.8	Stream Segmentation	
5.8.1	Ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums	
5.8.2	Ordered 1 byte segments, interleaved duplicate segments with null TCP control flags	
5.8.3	Ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream	
5.8.4	Ordered 1 byte segments, duplicate last packet	

Test ID	Description	Result
5.8.5	Ordered 2 byte segments, segment overlap (favor new)	
5.8.6	Ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers	
5.8.7	Out of order 1 byte segments	
5.8.8	Out of order 1 byte segments, interleaved duplicate segments with faked retransmits	
5.8.9	Ordered 1 byte segments, segment overlap (favor new)	
5.8.10	Out of order 1 byte segments, PAWS elimination (interleaved duplicate segments with older TCP time stamp options)	
5.8.11	Ordered 16 byte segments, segment overlap (favor new (Unix))	
5.9	RPC Fragmentation	
5.9.1	One-byte fragmentation (ONC)	
5.9.2	Two-byte fragmentation (ONC)	
5.9.3	All fragments, including Last Fragment (LF) will be sent in one TCP segment (ONC)	
5.9.4	All fragments except Last Fragment (LF) will be sent in one TCP segment. LF will be sent in separate TCP segment (ONC)	
5.9.5	One RPC fragment will be sent per TCP segment (ONC)	
5.9.6	One LF split over more than one TCP segment. In this case no RPC fragmentation is performed (ONC)	
5.9.7	Canvas Reference Implementation Level 1 (MS)	
5.9.8	Canvas Reference Implementation Level 2 (MS)	
5.9.9	Canvas Reference Implementation Level 3 (MS)	
5.9.10	Canvas Reference Implementation Level 4 (MS)	
5.9.11	Canvas Reference Implementation Level 5 (MS)	
5.9.12	Canvas Reference Implementation Level 6 (MS)	
5.9.13	Canvas Reference Implementation Level 7 (MS)	
5.9.14	Canvas Reference Implementation Level 8 (MS)	
5.9.15	Canvas Reference Implementation Level 9 (MS)	
5.9.16	Canvas Reference Implementation Level 10 (MS)	
5.10	URL Obfuscation	
5.10.1	URL encoding - Level 1 (minimal)	
5.10.2	URL encoding - Level 2	
5.10.3	URL encoding - Level 3	
5.10.4	URL encoding - Level 4	
5.10.5	URL encoding - Level 5	
5.10.6	URL encoding - Level 6	
5.10.7	URL encoding - Level 7	
5.10.8	URL encoding - Level 8 (extreme)	
5.10.9	Premature URL ending	

Test ID	Description	Result
5.10.10	Long URL	
5.10.11	Fake parameter	
5.10.12	TAB separation	
5.10.13	Case sensitivity	
5.10.14	Windows \ delimiter	
5.10.15	Session splicing	
5.11	HTML Obfuscation	
5.11.1	UTF-16 character set encoding (big-endian)	
5.11.2	UTF-16 character set encoding (little-endian)	
5.11.3	UTF-32 character set encoding (big-endian)	
5.11.4	UTF-32 character set encoding (little-endian)	
5.11.5	UTF-7 character set encoding	
5.11.6	Chunked encoding (random chunk size)	
5.11.7	Chunked encoding (fixed chunk size)	
5.11.8	Chunked encoding (chaffing)	
5.11.9	Compression (Deflate)	
5.11.10	Compression (Gzip)	
5.11.11	Base-64 Encoding	
5.11.12	Base-64 Encoding (shifting 1 bit)	
5.11.13	Base-64 Encoding (shifting 2 bits)	
5.11.14	Base-64 Encoding (chaffing)	
5.11.15	Combination UTF-7 + Gzip + Chunked encoding (random chunk size)	
5.12	FTP Evasion	
5.12.1	Inserting spaces in FTP command lines	
5.12.2	Inserting non-text Telnet opcodes - Level 1 (minimal)	
5.12.3	Inserting non-text Telnet opcodes - Level 2	
5.12.4	Inserting non-text Telnet opcodes - Level 3	
5.12.5	Inserting non-text Telnet opcodes - Level 4	
5.12.6	Inserting non-text Telnet opcodes - Level 5	
5.12.7	Inserting non-text Telnet opcodes - Level 6	
5.12.8	Inserting non-text Telnet opcodes - Level 7	
5.12.9	Inserting non-text Telnet opcodes - Level 8 (extreme)	
6	NIPS Performance	
6.1	Raw Packet Processing Performance (UDP Traffic)	
6.1.1	128 Byte Packets	
6.1.2	256 Byte Packets	
6.1.3	512 Byte Packets	
6.1.4	1024 Byte Packets	

Test ID	Description	Result
6.1.5	1514 Byte Packets	
6.2	Connection Dynamics – Concurrency and Connection Rates	
6.2.1	Theoretical Maximum Concurrent TCP Connections	
6.2.2	Theoretical Maximum Concurrent TCP Connections with Data	
6.2.3	Stateful Protection at Maximum Concurrent Connections	
6.2.4	Maximum TCP Connections Per Second	
6.2.5	Maximum HTTP Connections Per Second	
6.2.6	Maximum HTTP Transactions Per Second	
6.3	Behavior of the State Engine Under Load	
6.3.1	Attack Detection/Blocking - Normal Load	
6.3.2	State Preservation - Normal Load	
6.3.3	Pass Legitimate Traffic - Normal Load	
6.3.4	Attack Detection/Blocking - Maximum Exceeded	
6.3.5	State Preservation - Maximum Exceeded	
6.3.6	Pass Legitimate Traffic - Maximum Exceeded	
6.4	HTTP Capacity with No Transaction Delays	
6.4.1	2,500 Connections Per Second – 44 Kbyte Response	
6.4.2	5,000 Connections Per Second – 21 Kbyte Response	
6.4.3	10,000 Connections Per Second – 10 Kbyte Response	
6.4.4	20,000 Connections Per Second – 4.5 Kbyte Response	
6.4.5	40,000 Connections Per Second – 1.7 Kbyte Response	
6.5	HTTP Capacity with Transaction Delays	
6.5.1	21 Kbyte Response with Delay	
6.5.2	10 Kbyte Response with Delay	
6.6	“Real-World” Traffic	
6.6.1	“Real -World” Protocol Mix (Perimeter)	
6.6.2	“Real-World” Protocol Mix (Core)	
7	Management and Configuration Costs	
7.1	Ease-of-Use	
7.1.1	Initial Setup (Hours)	
7.1.2	Time Required for Upkeep (Hours per Year)	
7.1.3	Time Required to Tune (Hours per Year)	
7.2	Expected Costs	
7.2.1	Initial Purchase	
7.2.2	Ongoing Maintenance & Support (Annual)	
7.2.3	Installation Labor Cost (@\$75/hr)	
7.2.4	Management Labor Cost (per Year @\$75/hr)	
7.2.5	Tuning Labor Cost (per Year @\$75/hr)	

Test ID	Description	Result
7.3	Total Cost of Ownership	
7.3.1	Year 1	
7.3.2	Year 2	
7.3.3	Year 3	
7.3.4	3 Year Total Cost of Ownership	

5 SECURITY EFFECTIVENESS

This section verifies that the IPS 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 IPS 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 intrusion detection systems operate in detection-only mode, a Network IPS is required to block and log exploit attempts and hostile traffic. However, denial-of-service attacks are left to the dedicated NSS Labs Network Intrusion Prevention System testing track.

5.1 DETECTION ENGINE

While it is not possible to validate the entire signature set of any IPS, the NSS Labs testing provides a demonstration of effectiveness for the IPS to protect vulnerable assets from targeted threats and exploitation. This asset/target and threat based approach forms the basis from which network IPS security effectiveness is measured. (See NSS Labs' white paper *Intrusion Prevention Security Effectiveness*, available at www.nsslabs.com).

The NSS Labs 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, NSS Labs categorizes threats and exploits 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 network IPS).

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 IPS, the service could have crashed because of the residual communications resulting in a fault impact on the service or operating system.

5.3 TARGET TYPE

The following list of web target types is represented in NSS Labs' live exploit test. Protection capabilities are indicated as percentages.

Web Server	Web Browser
ActiveX	JavaScript
Browser Plug-ins/Add-ons	

5.4 COVERAGE BY RESULT

The following results of exploitation are represented in NSS Labs' live exploit test. Protection capabilities are indicated as percentages.

5.4.1 ARBITRARY CODE EXECUTION

A software bug that allows an attacker to execute any commands of the attacker's choice on a target machine or in a target process

5.4.2 BUFFER OVERFLOW

The exploitation of a software bug due to improperly establishing memory bounds allows an attacker to overwrite adjacent memory and execute a command.

5.4.3 CODE INJECTION

The exploitation of a software bug what allows the processing of invalid data within a program. Code injection can be used by an attacker to introduce code into a computer program to change the course of execution.

5.4.4 CROSS-SITE SCRIPT

The exploitation of a web application which enables attackers to insert malicious script into web pages which is then viewed by other users.

5.4.5 DIRECTORY TRAVERSAL

The exploitation of a lack of security in an application (as opposed to exploiting a bug in the code) which allows user supplied input with characters representing "traverse to parent directory" to be passed through to the file APIs. The goal of this attack is to order an application to access a file or executable that is not intended to be accessible.

5.4.6 PRIVILEGE ESCALATION

This exploit type allows an attacker to gain access to resources which would not normally have been available.

5.5 COVERAGE BY VENDOR

The following list of vendors is represented in NSS Labs' live exploit test. Protection capabilities are indicated as percentages.

- 3Com
- Alt-N
- Apple
- Avast
- BitDefender
- CA
- Citrix
- EMC
- GNU
- HP
- IPSwitch
- Kaspersky
- lighttpd
- Adobe
- Apache
- Atrium
- BEA
- Borland
- Cisco
- ClamAV
- Facebook
- Google
- IBM
- ISC
- LanDesk
- Linux

- Macromedia
- Mailenable
- Mercury
- MIT
- Mplayer
- MySQL
- Novell
- OpenLDAP
- OpenSSH
- Oracle
- Panda
- Samba
- Snort
- SpamAssassin
- Sun Microsystems
- Trend Micro
- UltraVNC
- VideoLan
- WinAmp
- Winzip
- MacroVision
- McAfee
- Microsoft
- Mozilla
- Multiple Vendors
- NOD32
- Nullsoft
- OpenOffice
- OpenSSL
- Other Misc
- RealNetworks
- SAP
- Sophos
- Squid
- Symantec
- Trillian
- Veritas
- VMWare
- WinFTP
- Yahoo

5.6 EVASION

Cyber-criminals can modify basic attacks to evade detection in a number of ways. If an IPS fails to detect a single form of evasion, any exploit can pass through the device, rendering it ineffective. NSS Labs verifies that the IPS is capable of detecting and blocking basic exploits when subjected to varying common evasion techniques. Further, the DUT is expected to successfully “decoded” the evasion to provide an accurate alert relating to the original exploit, rather than alerting purely on anomalous traffic detected as a result of the evasion technique itself.

5.6.1 UNMODIFIED EXPLOIT VALIDATION

A number of common exploits are executed across the IPS 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 Labs 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 Labs exploit library in advance of the test.

5.7 PACKET FRAGMENTATION

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

- Ordered 8 byte fragments
- Ordered 24 byte fragments
- Out of order 8 byte fragments
- Ordered 8 byte fragments, duplicate last packet
- Out of order 8 byte fragments, duplicate last packet
- Ordered 8 byte fragments, reorder fragments in reverse
- Ordered 16 byte fragments, fragment overlap (favor new)
- Ordered 16 byte fragments, fragment overlap (favor old)
- Out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery

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

5.8 STREAM SEGMENTATION

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

- Ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums
- Ordered 1 byte segments, interleaved duplicate segments with null TCP control flags
- Ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream
- Ordered 1 byte segments, duplicate last packet
- Ordered 2 byte segments, segment overlap (favor new)
- Ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers
- Out of order 1 byte segments
- Out of order 1 byte segments, interleaved duplicate segments with faked retransmits
- Ordered 1 byte segments, segment overlap (favor new)
- Out of order 1 byte segments, PAWS elimination (interleaved duplicate segments with older TCP timestamp options)
- Ordered 16 byte segments, segment overlap (favor new (Unix))

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

5.9 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 IPS:

- One-byte fragmentation (ONC)
- Two-byte fragmentation (ONC)
- All fragments, including Last Fragment (LF) will be sent in one TCP segment (ONC)
- All fragments except Last Fragment (LF) will be sent in one TCP segment. LF will be sent in separate TCP segment (ONC)
- One RPC fragment will be sent per TCP segment (ONC)
- One LF split over more than one TCP segment (in this case, no RPC fragmentation is performed (ONC))
- Canvas Reference Implementation Level 1 (MS)
- Canvas Reference Implementation Level 2 (MS)
- Canvas Reference Implementation Level 3 (MS)

- Canvas Reference Implementation Level 4 (MS)
- Canvas Reference Implementation Level 5 (MS)
- Canvas Reference Implementation Level 6 (MS)
- Canvas Reference Implementation Level 7 (MS)
- Canvas Reference Implementation Level 8 (MS)
- Canvas Reference Implementation Level 9 (MS)
- Canvas Reference Implementation Level 10 (MS)

5.10 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.

- URL encoding - Level 1 (minimal)
- URL encoding - Level 2
- URL encoding - Level 3
- URL encoding - Level 4
- URL encoding - Level 5
- URL encoding - Level 6
- URL encoding - Level 7
- URL encoding - Level 8 (extreme)
- Premature URL ending
- Long URL
- Fake parameter
- TAB separation
- Case sensitivity
- Windows \ delimiter
- Session splicing

5.11 HTML OBFUSCATION

Recognizing malicious HTML documents is becoming increasingly important when protecting the enterprise. Malicious HTML documents exploit flaws in common web browsers, browser plug-ins, and add-ons to gain control of the client system and silently install malware such as Trojans, rootkits, and key loggers.

Therefore, it is becoming increasingly important that security products charged with protecting end systems must correctly interpret HTML documents. Many security products use simple pattern matching systems with very little semantic or syntactic understanding of the data they are analyzing. This leaves them vulnerable to evasion through use of redundant, but equivalent, alternative representations of malicious documents.

This test suite uses a number of malicious HTML documents which are transferred from server to client through the DUT. Each malicious HTML document is served with a different form of obfuscation, as follows:

- UTF-16 character set encoding (big-endian)
- UTF-16 character set encoding (little-endian)
- UTF-32 character set encoding (big-endian)
- UTF-32 character set encoding (little-endian)
- UTF-7 character set encoding
- Chunked encoding (random chunk size)
- Chunked encoding (fixed 8 byte chunk size)
- Chunked encoding (chaffing / arbitrary numbers inserted between chunks)
- Compression (Deflate)
- Compression (Gzip)
- Combination: UTF-7 encoding + Gzip compression + chunked encoding (random chunk sizes)

The UTF-16 character set specifies a 2-byte sequence for most characters and a 4-byte sequence for the others (a small percentage). Recoding an HTML document in UTF-16 significantly changes its appearance. A document that contains just the ASCII subset of characters will appear to have a null byte between every one of the original characters. There are also two different forms of the UTF-16 encoding depending on whether the null high byte comes first (big-endian) or second (little-endian) - this test uses big-endian byte ordering

The UTF-32 character set specifies a 4-byte sequence. Like the UTF-16 character set encoding there are two variations -big-endian and little-endian and this test case uses big-endian byte ordering.

The UTF-7 character set encodes most ASCII characters as themselves. However, in addition to recoding non-English characters as other encodings do, it also recodes many punctuation symbols, including many of the symbols that are important to the HTML specification. Therefore, recoding an HTML document in UTF-7 significantly changes its appearance

Chunked encoding allows the server to break a document into smaller chunks and transmit them individually. The server needs only to specify the size of each chunk before it is transmitted and then indicate when the last chunk has been transmitted. Since chunked encoding intersperses arbitrary numbers (chunk sizes) with the elements of the original document, it can be used to greatly change the appearance of the original document as observed "on the wire". In addition, the server can choose to break the document into chunks at arbitrary points. This makes it difficult for simple pattern matching systems to reliably identify the original HTML document from the raw data on the network.

Per RFC 2616, the HTTP protocol allows the client to request and the server to use several compression methods. These compression methods not only improve performance in many circumstances, they completely change the characteristic size and appearance of HTML documents. Furthermore, small changes in the original document, can greatly change the final appearance of the compressed document. This property of these algorithms could be used to obfuscate hostile content for the purpose of evading detection. The deflate compression method is a Lempel-Ziv coding (LZ77), specified in RFC 1951. The gzip compression method is specified in RFC 1952.

For each of the above, it is verified that a standard Web browser (such as Internet Explorer) is capable of rendering the results of the evasion.

5.12 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):

- Inserting spaces in FTP command lines
- Inserting non-text Telnet opcodes - Level 1 (minimal)
- Inserting non-text Telnet opcodes - Level 2
- Inserting non-text Telnet opcodes - Level 3
- Inserting non-text Telnet opcodes - Level 4
- Inserting non-text Telnet opcodes - Level 5
- Inserting non-text Telnet opcodes - Level 6
- Inserting non-text Telnet opcodes - Level 7
- Inserting non-text Telnet opcodes - Level 8 (extreme)

6 PERFORMANCE

This section measures the performance of the IPS 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 IPS 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 IPS 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 1 Gbps connections will be made from the external to internal switches via the IPS, subject to a minimum of one in-line port pair per Gigabit of throughput. Thus, an 8 Gbps device with only four port pairs will be limited to 4 Gbps. The minimum number of port pairs will be connected to support the claimed maximum bandwidth of the IPS. An 8 Gbps device with 10 port pairs will be deployed with eight, 1 Gbps connections.

Attacks are launched through the IPS against protected hosts with zero background traffic to ensure the IPS is capable of detecting the baseline attacks. Once that has been established, increasing levels of varying types of background traffic are generated through the IPS to determine the point at which the IPS 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 IPS will fail the overall test immediately.

6.1 RAW PACKET PROCESSING PERFORMANCE (UDP TRAFFIC)

This test uses UDP packets of varying sizes generated by both BreakingPoint Systems and 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 IPS.

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 IPS, 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 IPS, 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 IPS.

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 IPS.

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 IPS. 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 IPS.

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 IPS. This test has been included mainly to demonstrate how easy it is to achieve good results using large packets. Readers should use caution when taking into consideration those test results that only quote performance figures using similar packet sizes.

6.2 CONNECTION DYNAMICS – CONCURRENCY AND CONNECTION RATES

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

- **Unsuccessful HTTP transactions/SMTP sessions** – normally, there should be zero unsuccessful transactions. Once these appear, it is an indication that excessive latency within the IPS 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 IPS 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 IPS. 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 the one described above in Section 6.2.1, except that once the maximum number of concurrent connections has been established, 1 GB of data is transmitted across them in 21 KB segments. This ensures that the IPS 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 the one described above in Section 6.2.1, but is designed to verify the maximum concurrent TCP connections on which the vendor claims the IPS 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 IPS with 1 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 IPS, ramped slowly to determine the exact point of failure. Each session is opened normally, one byte of data passed through the IPS, 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 IPS 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 process ensures that all TCP connections are closed immediately once the request is satisfied, thus, any concurrent TCP connections will be caused purely as a result of latency within the IPS. 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 IPS 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 10 HTTP requests, and close the connection. This ensures that TCP connections remain open until all 10 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 IPS 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 IPS is still capable of detecting and blocking freshly-launched exploits, as well as confirming that the device does not block legitimate traffic .

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% of the maximum determined in Section 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% of the maximum determined in Section 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% of the maximum determined in Section 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 Section 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 Section 6.2.1. The method of execution is identical to that described in Section 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. **Note:** 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. This test provides an excellent representation of a live network (albeit one biased towards HTTP traffic) at various network loads.

6.4.1 44 KBYTE HTTP RESPONSE

Maximum 2,500 new connections per second per Gigabit of traffic with a 44 KB 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 IPS. With relatively low connection rates and large packet sizes, all sensors should be capable of performing well throughout this test.

6.4.2 21 KBYTE HTTP RESPONSE

Maximum 5,000 new connections per second per Gigabit of traffic with a 21 KB 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 IPS. 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 KBYTE HTTP RESPONSE

Maximum 10,000 new connections per second per Gigabit of traffic with a 10 KB 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 IPS. 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 **4.5 KBYTE HTTP RESPONSE**

Maximum 20,000 new connections per second per Gigabit of traffic with a 4.5 KB 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 IPS. With small packet sizes and extremely high connection rates, this is an extreme test for any sensor.

6.4.5 **1.7 KBYTE HTTP RESPONSE**

Maximum 40,000 new connections per second per Gigabit of traffic with a 1.7 KB 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 IPS. With small packet sizes and extremely high connection rates, this is an extreme test for any sensor.

6.5 **HTTP CAPACITY WITH TRANSACTION DELAYS**

Typical user behavior introduces delays in between requests and responses, e.g. as users read web pages and decide which links to click next. This next set of tests is identical to the previous set except that these include a 10second 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 **21 KBYTE HTTP RESPONSE 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 IPS. 10 second transaction delay resulting in an additional 50,000 open connections over the test described in Section 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 **10 KBYTE HTTP 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 IPS. 10 second transaction delay resulting in an additional 100,000 open connections over the test described in Section 6.4.3. With large 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.

6.6 **“REAL-WORLD” TRAFFIC**

Where 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.

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 IPS comprising the following protocol mix typically 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 (>300 kb) 4%
- SSH 4%
- AOL IM 3%
- SIP/RTP 1%
- BitTorrent 1%

For this test and the one described in Section 6.6.2, HTTP traffic comprises genuine transactions and web pages from real websites such as Google, Yahoo, MSN, and NSS Labs, including small (< 50KB) and large (>300 KB) 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 IPS. 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 IPS engine, this test is a good approximation of a heavily-used production network. All sensors should be capable of performing well throughout this test (and the one described below in Section 6.6.2).

6.6.2 "REAL-WORLD" PROTOCOL MIX (CORE)

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

- HTTP text 24%
- SMB File transfer 14%
- HTTP Images (<50 kb) 12%
- SMTP 12%
- PostgreSQL 10%

- DNS 6%
- DCERPC 4%
- FTP 3%
- SMB NULL 3%
- HTTP Video 2%
- HTTP Audio 2%
- HTTP Images (>300 kb) 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 IPS. Maximum of 6,000 open connections during the test.

7 MANAGEMENT AND CONFIGURATION COSTS

Organizations should be concerned with the ongoing, amortized cost of operating security products. This section evaluates the costs associated with the purchase, installation, and ongoing management of the IPS.

7.1 EASE-OF-USE

7.1.1 INITIAL SETUP (HOURS)

The initial setup costs include those associated with time required to install and configure the IPS.

7.1.2 TIME REQUIRED FOR UPKEEP (HOURS PER YEAR)

The time required to keep the device in good health. This includes applying patches, OS updates, exporting and archiving logs, generating reports, and backing up the system.

7.1.3 TIME REQUIRED TO TUNE (HOURS PER YEAR)

The time required to keep create, deploy, and maintain the optimal security policy. This includes downloading signatures/filters, defining updated policies based upon those new signatures, and eliminating all false positives.

7.2 EXPECTED COSTS

7.2.1 INITIAL PURCHASE

7.2.2 ONGOING MAINTENANCE & SUPPORT (ANNUAL)

7.2.3 INSTALLATION LABOR COST (@\$75/HR)

7.2.4 MANAGEMENT LABOR COST (PER YEAR @\$75/HR)

7.2.5 TUNING LABOR COST (PER YEAR @\$75/HR)

7.3 TOTAL COST OF OWNERSHIP

7.3.1 YEAR 1

7.3.2 YEAR 2

7.3.3 YEAR 3

7.3.4 THREE-YEAR TOTAL COST OF OWNERSHIP

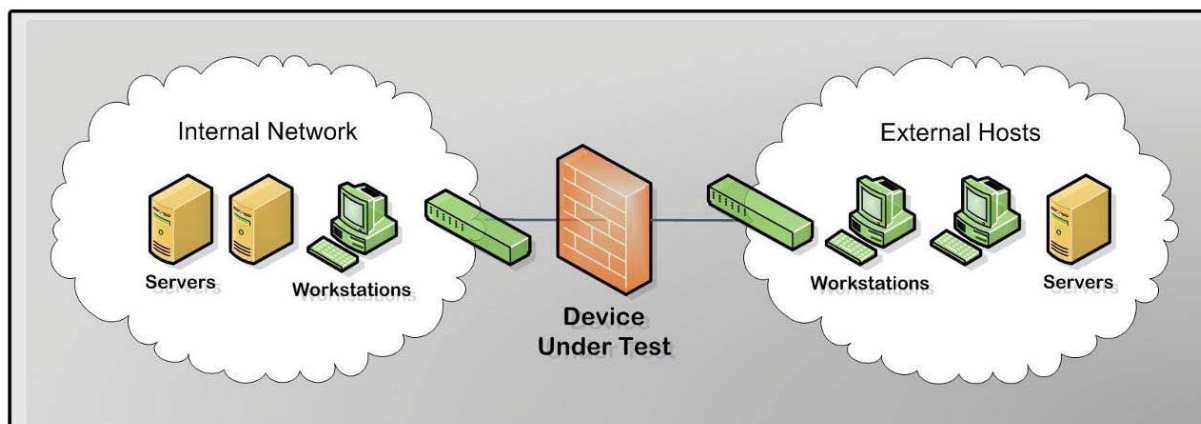
APPENDIX A: NETWORK IPS TEST ENVIRONMENT

The aim of this procedure is to provide a thorough test of all the main components of an in-line IPS 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 IPS is configured for the use-case appropriate to the target deployment environment.

Traffic generation equipment—such as the hosts generating exploits, BreakingPoint and Smartbits transmit ports—is connected to the “external” network, while the “receiving” equipment—such as the vulnerable hosts for the exploits, BreakingPoint and 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 is 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 1 Gbps.

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: SPECIAL THANKS

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


BreakingPointTM

Find it before they do.TM

