# Top Layer Attack Mitigator IPS 2400 V2.1

## Technical Evaluation

**An NSS Group Report**

First published January 2004 (Version 1.0)

Published by The NSS Group
Mas la Carrière, Route de Ganges
30440 Sumène, France

Tel : +33 (0)4 67 81 49 11
E-mail : info@nss.co.uk
Internet : http://www.nss.co.uk



The NSS Group Limited is registered in England & Wales, Reg No. 3233843
Registered Office: Montagu House, 81 High Street, Huntingdon, Cambs, PE29 3NY, England
Tel +44 (0)7005 802 953

# TABLE OF CONTENTS

# TABLE OF FIGURES

# The NSS Group

The NSS Group is Europe's foremost independent security testing facility.

Based in the UK with separate security and network infrastructure testing facilities in the South of France, The NSS Group offers a range of specialist IT, networking and security-related services to vendors and end-user organisations throughout Europe and the United States.

The Group consists of two wholly-owned subsidiaries :

- *NSS Network Testing Laboratories*
- *Network Security Services*

**NSS Network Testing Laboratories** are available to vendors and end-users for fully independent testing of networking, communications and security hardware and software.

NSS Network Testing Laboratories also operates certification schemes for vendors and certification bodies, and currently provides certification of firewalls, VPN's, crypto products and PKI products.

Output from the labs, including detailed research reports, articles and white papers on the latest network and security technologies, are made available on the NSS web site at http://www.nss.co.uk

**Network Security Services** provides a range of security-related services to vendors and end-users including security policy definition, IDS, firewall and VPN implementation, network security auditing and analysis, and penetration testing.

# Foreword

The NSS Group is pleased to present the results of the first comprehensive *Intrusion Prevention System* (IPS) test of its kind.

This exhaustive review will give readers a complete perspective of the capabilities, maturity and suitability for immediate deployment of each of the products tested. The NSS Group established this test as IPS products are being actively deployed as a new layer in defence-in-depth security architectures.

Contrary to recent analyst claims, we do not believe that "IDS is dead" or that "IPS is stillborn". So-called "*deep inspection firewalls*" **may** be where the industry is heading in the long term, but they are simply not ready for prime-time deployments at this point in time. Until they are, security administrators need to make the best use of the technology that **is** available, and for now that means a combination of firewalls, in-line intrusion prevention devices, and intrusion detection systems.

Please note that we fully recognise that each of the above product-types could be considered as "intrusion prevention" systems in some respect, along with Anti Virus gateways, desktop firewalls, and other products designed to "prevent" malicious activity on a network or individual host. However, we also believe that the marketing terms for each of these products are well established, and that the grounds for creating a new market segment - referred to as *Intrusion Prevention Systems* (IPS) - specifically for those products which are evaluated as part of this report is valid. We have defined what **we** consider to be IPS (both host- and network-based) in the introductory text to this report.

You might not like the marketing hype, but the time for quibbling over the terminology is over - it is now time to get down to the serious issue of evaluating the technology behind it.

The NSS IPS Group Test evaluates the performance, reliability, security effectiveness, and usability of both Network IPS and Host IPS products. The test consists of seven sections within three primary areas: *performance and reliability, security accuracy,* and *usability*.

Overall, the suite contains over **750 individual tests**, many of which are run multiple times, to provide the most thorough and complete evaluation of IPS products available anywhere today.

We believe that our IPS test methodology will become the *de facto* standard for testing in-line intrusion prevention devices, and the *NSS Approved* logo an essential item on the list of requirements when purchasing these products.

We also believe that this report is essential reading for anyone considering deploying Intrusion Prevention Systems in their networks, either in a test or live situation, and we hope that you find it both informative and useful in making your purchasing decisions.

*Bob Walder*

# INTRODUCTION

In a recent survey commissioned by VanDyke Software, some 66 per cent of the companies who responded said that they perceive system penetration to be the largest threat to their enterprises.

The survey revealed that the top eight threats experienced by those surveyed were *viruses* (78 per cent of respondents), *system penetration* (50 per cent), *DoS* (40 per cent), *insider abuse* (29 per cent), *spoofing* (28 per cent), *data/network sabotage* (20 per cent), and *unauthorised insider access* (16 per cent).

Although 86 per cent of respondents use firewalls (a disturbingly **low** figure in this day and age, to be honest!), it is apparent that firewalls are not always effective against many intrusion attempts. The average firewall is designed to deny clearly suspicious traffic - such as an attempt to telnet to a device when corporate security policy forbids telnet access completely - but is also designed to allow some traffic through - Web traffic to an internal Web server, for example.

The problem is, that many exploits attempt to take advantage of weaknesses in the very protocols that **are** allowed through our perimeter firewalls, and once the Web server has been compromised, this can often be used as a springboard to launch additional attacks on other internal servers. Once a “rootkit” or “back door” has been installed on a server, the hacker has ensured that he will have unfettered access to that machine at any point in the future.

Firewalls are also typically employed only at the network perimeter. However, many attacks, intentional or otherwise, are launched from within an organisation. Virtual private networks, laptops, and wireless networks all provide access to the internal network that often bypasses the firewall. Intrusion detection systems may be effective at detecting suspicious activity, but do not provide *protection* against attacks. Recent worms such as Slammer and Blaster have such fast propagation speeds that by the time an alert is generated, the damage is done and spreading fast.

## Intrusion Prevention Systems (IPS)

The inadequacies inherent in current defences has driven the development of a new breed of security products known as *Intrusion Prevention Systems* (IPS). This is a term which has provoked some controversy in the industry since some firewall and IDS vendors think it has been “hijacked” and used as a marketing term rather than as a description for any kind of new technology.

Whilst it is true that firewalls, routers, IDS devices and even AV gateways all have intrusion prevention technology included in some form, we believe that there are sufficient grounds to create a new market sector for true *Intrusion Prevention Systems*.

These systems are proactive defence mechanisms designed to detect malicious packets within normal network traffic (something that the current breed of firewalls do not actually do, for example) and stop intrusions dead, blocking the offending traffic automatically before it does any damage rather than simply raising an alert as, or after, the malicious payload has been delivered.

Within the IPS market place, there are two main categories of product: *Host IPS* and *Network IPS*.

## Host IPS (HIPS)

As with Host IDS systems, the Host IPS relies on agents installed directly on the system being protected. It binds closely with the operating system kernel and services, monitoring and intercepting system calls to the kernel or APIs in order to prevent attacks as well as log them.

It may also monitor data streams and the environment specific to a particular application (file locations and Registry settings for a Web server, for example) in order to protect that application from generic attacks for which no “signature” yet exists.

One potential disadvantage with this approach is that, given the necessarily tight integration with the host operating system, future OS upgrades could cause problems.

Since a Host IPS agent intercepts all requests to the system it protects, it has certain prerequisites - it must be very reliable, must not negatively impact performance, and must not block legitimate traffic. Any HIPS that does not meet these minimum requirements should never be installed in a host, no matter how effectively it blocks attacks.

## Network IPS (NIPS)

The Network IPS combines features of a standard IDS, an IPS and a firewall, and is sometimes known as an *In-line IDS* or *Gateway IDS (GIDS)*. The next-generation firewall - the *deep inspection firewall* - also exhibits a similar feature set, though we do not believe that the deep inspection firewall is ready for mainstream deployment just yet.

As with a typical firewall, the NIPS has at least two network interfaces, one designated as *internal* and one as *external*. As packets appear at the either interface they are passed to the detection engine, at which point the IPS device functions much as any IDS would in determining whether or not the packet being examined poses a threat.

However, if it should detect a malicious packet, in addition to raising an alert, it will discard the packet and mark that flow as bad. As the remaining packets that make up that particular TCP session arrive at the IPS device, they are discarded immediately.

Legitimate packets are passed through to the second interface and on to their intended destination. A useful side effect of some NIPS products is that as a matter of course - in fact as part of the initial detection process - they will provide “*packet scrubbing*” functionality to remove protocol inconsistencies resulting from varying interpretations of the TCP/IP specification (or intentional packet manipulation).

Thus any fragmented packets, out-of-order packets, or packets with overlapping IP fragments will be re-ordered and “cleaned up” before being passed to the destination host, and illegal packets can be dropped completely.

One thing to watch out for - don't let the "reactive" IDS vendors kid you into believing that they have *intrusion prevention* capabilities just because they can send TCP reset commands or re-configure a firewall when they detect an attack (a worrying piece of FUD that we have noticed in some IDS marketing literature recently).

The problem here is that unless the attacker is operating on a 2400 baud modem, the likelihood is that by the time the IDS has detected the offending packet, raised an alert, and transmitted the TCP Resets - and especially by the time the two ends of the connection have received the Reset packets and acted on them (or the firewall or router has had time to activate new rules to block the remainder of the flow) - the payload of the exploit has long since been delivered….. *game over*! Our guess is that there are not many crackers using 2400 baud modems these days….

A true IPS device, however, is sitting in-line - **all** the packets have to pass through it. Therefore, as soon as a suspicious packet has been detected - and **before** it is passed to the internal interface and on to the protected network, it can be dropped. Not only that, but now that flow has been flagged as suspicious, **all** subsequent packets that are part of that session can also be dropped with very little additional processing. Oh, and for good measure, some products are also capable of sending *TCP Resets* or *ICMP Unreachable* messages to the attacking host.

## Implementation Challenges

There are a number of challenges to the implementation of an IPS device that do not have to be faced when deploying passive-mode IDS products. These challenges all stem from the fact that the IPS device is designed to work in-line, presenting a potential choke point and single point of failure.

If a passive IDS fails, the worst that can happen is that some attempted attacks may go undetected. If an in-line device fails, however, it can seriously impact the performance of the network. Perhaps latency rises to unacceptable values, or perhaps the device fails closed, in which case you have a self-inflicted Denial of Service condition on your hands. On the bright side, there will be no attacks getting through! But that is of little consolation if none of your customers can reach your e-commerce site.

Even if the IPS device does not fail altogether, it still has the potential to act as a bottleneck, increasing latency and reducing throughput as it struggles to keep up with up to a Gigabit or more of network traffic. Devices using off-the-shelf hardware will certainly struggle to keep up with a heavily loaded Gigabit network, especially if there is a substantial signature set loaded, and this could be a major concern for both the network administrator - who could see his carefully crafted network response times go through the roof when a poorly designed IPS device is placed in-line - as well as the security administrator, who will have to fight tooth and nail to have the network administrator allow him to place this unknown quantity amongst his high performance routers and switches.

As an integral element of the network fabric, the Network IPS device must perform much like a network switch. It must meet stringent network performance and reliability requirements as a prerequisite to deployment, since very few customers are willing to sacrifice network performance and reliability for security. A NIPS that slows down traffic, stops good traffic, or crashes the network is of little use.

Dropped packets are also an issue, since if even one of those dropped packets is one of those used in the exploit data stream it is possible that the entire exploit could be missed. Most high-end IPS vendors will get around this problem by using custom hardware, populated with advanced FPGAs and ASICs - indeed, it is necessary to design the product to operate as much as a switch as an intrusion detection and prevention device.

It is very difficult for any security administrator to be able to characterise the traffic on his network with a high degree of accuracy. What is the average bandwidth? What are the peaks? Is the traffic mainly one protocol or a mix? What is the average packet size and level of new connections established every second - both critical parameters that can have detrimental effects on some IDS/IPS engines? If your IPS hardware is operating “on the edge”, all of these are questions that need to be answered as accurately as possible in order to prevent performance degradation.

Another potential problem is the good old *false positive*. The bane of the security administrator’s life (apart from the script kiddie, of course!), the false positive rears its ugly head when an exploit signature is not crafted carefully enough, such that legitimate traffic can cause it to fire accidentally. Whilst merely annoying in a passive IDS device, consuming time and effort on the part of the security administrator, the results can be far more serious and far reaching in an in-line IPS appliance.

Once again, the result is a self-inflicted Denial of Service condition, as the IPS device first drops the “offending” packet, and then potentially blocks the entire data flow from the suspected hacker. If the traffic that triggered the false positive alert was part of a customer order, you can bet that the customer will not wait around for long as his entire session is torn down and all subsequent attempts to reconnect to your e-commerce site (if he decides to bother retrying at all, that is) are blocked by the well-meaning IPS.

Another potential problem with any Gigabit IPS/IDS product is, by its very nature and capabilities, the amount of alert data it is likely to generate. On such a busy network, how many alerts will be generated in one working day? Or even one hour? Even with relatively low alert rates of ten per second, you are talking about 36,000 alerts every hour. That is 864,000 alerts each and every day. The ability to tune the signature set accurately is essential in order to keep the number of alerts to an absolute minimum. Once the alerts have been raised, however, it then becomes essential to be able to process them effectively. Advanced alert handling and forensic analysis capabilities - including detailed exploit information and the ability to examine packet contents and data streams - can make or break a Gigabit IDS/IPS product.

Of course, one point in favour of IPS when compared with IDS is that because it is designed to prevent the attacks rather than just detect and log them, the burden of examining and investigating the alerts - and especially the problem of rectifying damage done by successful exploits - is reduced considerably.

## Requirements for effective prevention

Having pointed out the potential pitfalls facing anyone deploying these devices, what features are we looking for that will help us to avoid such problems?

- **In-line operation -** only by operating in-line can an IPS device perform true protection, discarding all suspect packets immediately and blocking the remainder of that flow
- **Reliability and availability -** should an in-line device fail, it has the potential to close a vital network path and thus, once again, cause a DoS condition. An extremely low failure rate is thus very important in order to maximise up-time, and if the worst should happen, the device should provide the option to fail open or support fail-over to another sensor operating in a fail-over group (see below). In addition, to reduce downtime for signature and protocol coverage updates, an IPS must support the ability to receive these updates without requiring a device re-boot. When operating inline, sensors rebooting across the enterprise effectively translate into network downtime for the duration of the reboot
- **Resilience -** as mentioned above, the very minimum that an IPS device should offer in the way of High Availability is to fail open in the case of system failure or power loss (some environments may prefer this default condition to be “fail closed” as with a typical firewall, however - the most flexible products will allow this to be user-configurable). Active-Active stateful fail-over with cooperating in-line sensors in a fail-over group will ensure that the IPS device does not become a single point of failure in a critical network deployment
- **Low latency -** when a device is placed in-line, it is essential that its impact on overall network performance is minimal. Packets should be processed quickly enough such that the overall latency of the device is as close as possible to that offered by a layer 2/3 device such as a switch, and no more than a typical layer 4 device such as a firewall or load-balancer.
- **High performance -** packet processing rates must be at the rated speed of the device under real-life traffic conditions, and the device must meet the stated performance with all signatures enabled. Headroom should be built into the performance capabilities to enable the device to handle any increases in size of signature packs that may occur over the next three years. Ideally, the detection engine should be designed in such a way that the number “signatures” (or “checks”) loaded does not affect the overall performance of the device.
- **Unquestionable detection accuracy -** it is imperative that the quality of the signatures is beyond question, since false positives can lead to a Denial of Service condition. The user MUST be able to trust that the IDS is blocking only the user selected malicious traffic. New signatures should be made available on a regular basis, and applying them should be quick (applied to all sensors in one operation via a central console) and seamless (no sensor reboot required)
- **Fine-grained granularity and control -** fine grained granularity is required in terms of deciding exactly which malicious traffic is blocked. The ability to specify traffic to be blocked by attack, by policy, or right down to individual host level is vital. In addition, it may be necessary to only alert on suspicious traffic for further analysis and investigation
- **Advanced alert handling and forensic analysis capabilities -** once the alerts have been raised at the sensor and passed to a central console, someone has to examine them, correlate them where necessary, investigate them, and eventually decide on an action. The capabilities offered by the console in terms of alert viewing (real time and historic) and reporting are key in determining the effectiveness of the IPS product.

# The NSS Intrusion Prevention Group Test

The NSS Group has conducted the first comprehensive IPS test of its kind. This exhaustive review will give readers a complete perspective of the capabilities, maturity and suitability of the products tested for their particular needs.

As part of its extensive IPS test methodology (see section on *Testing Methodology* later in this report for detailed methodology) The NSS Group subjects each product to a brutal battery of tests that verify the stability and performance of each IPS tested, determine the accuracy of its security coverage, and ensure that the device will not block legitimate traffic.

**If a particular IPS has been designated as *NSS Approved*, customers can be confident that the device will not significantly impact network/host performance, cause network/host crashes, or otherwise block legitimate traffic.**

To assess the complex matrix of IPS performance and security requirements, the NSS Group has developed a specialised lab environment that is able to exercise every facet of an IPS product. The test suite contains over 750 individual tests that evaluate IPS products in three main areas: *performance and reliability, security accuracy,* and *usability*. This thorough review should give readers a complete perspective of the capabilities, maturity and suitability of the products tested for their particular needs.

## Performance

Any IPS is expected to be reliable (not crash), to never block legitimate traffic, and to not unduly affect network or host system performance.

The latency and throughput of a network IPS (NIPS) device must be on a par with other equipment in the network on which it is deployed, and in this respect, an in-line NIPS must strive to perform much more like a switch than a typical passive security device, especially when it is necessary to install more than one NIPS in the same data path.

### Detection/Blocking Performance Under Load

This group of tests verifies that the IPS does not adversely impact legitimate traffic, even when new TCP connections are being created rapidly. We also verify that the sensor is capable of detecting and blocking exploits when subjected to increasing loads of background traffic up to the maximum bandwidth supported as claimed by the vendor. An IPS that misses attacks under load can be evaded. An IPS that adversely affects legitimate background traffic will not stay in-line for long.

A fixed number of exploits are launched with zero background traffic to ensure the sensor is capable of detecting our baseline attacks. Once that has been established, increasing levels of varying types of background traffic are generated **through** the IPS device in order to determine the point at which the sensor begins to miss attacks.

All tests are repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic (or up to the maximum rated throughput of the device in 25 per cent increments should this be less than 1Gbps).

The test is conducted with UDP, HTTP, and mixed-protocol traffic and includes packet rates up to 1.48 million packets per second and connection rates up to 20,000 connections per second.

### Latency & User Response Times

In any network environment latency is important. Latency may impose an upper bound on throughput and it also has an impact on interactive applications, thus affecting user response time. As such, it is important to understand the impact of latency introduced by a NIPS and to determine the maximum acceptable delay, which will be different for each network.

There is a direct relationship between latency introduced by a networking device and the maximum throughput allowed by that device on a single TCP connection. There is a critical value for the *round trip time* (RTT) of a packet in each network, and if the latency is below this critical value, TCP throughput will be unaffected - instead, it is the line speed of the underlying network which becomes the bottleneck. Above this critical value, however, TCP throughput is negatively impacted.

To be specific, the maximum throughput achievable for any given TCP connection in a zero loss network is expressed as:

*throughput = window / RTT*

where *window* is the maximum TCP window size (64 Kbytes by default) and RTT is the round trip time in the network. This equation tells us that the throughput of a TCP connection is inversely proportional to network latency (note that this is TCP throughput for *one* connection - the aggregate bandwidth is not affected by latency). In other words, if you double latency, you halve throughput.

Consider adding a NIPS in an internal Gigabit network where the RTT is 200 microseconds. The critical value for RTT in a Gigabit network is 500 microseconds (below which it may no longer be possible to achieve 1Gbps of throughput), which means the NIPS can add a maximum of 300 microseconds to the RTT without affecting the network. In this particular case, therefore, for an internal, high speed deployment, the administrator may determine that his chosen IPS device needs to be capable of sub-300 microsecond latency under normal traffic loads.

Of course, the latency of an IPS device may vary significantly based on packet size, complexity of the protocol, presence of attack traffic, or simply the makeup of the normal traffic passing through it. For example, Gigabit segments, will rarely carry only a single TCP connection. Rather, a saturated Gigabit segment could be supporting hundreds, if not thousands of TCP connections, and this multiplexing eases the impact of latency on the overall throughput on the segment.

Although each of these connections carries only a fraction of the total throughput, a few connections tend to dominate. The maximum latency for a NIPS is then determined by the utilisation of the fastest connection. For example, in a Gigabit Ethernet segment carrying 10,000 TCP connections the fastest connection might have a throughput of 250Mbps. In this case, the critical value for round trip latency is as high as 2 milliseconds.

Assuming the latency without the NIPS is 300 microseconds, an administrator may therefore determine that his chosen NIPS device must be capable of 1700 microsecond round trip latency (850 microseconds in each direction).

Such critical value calculations are important when TCP connections achieve maximum throughput, which is true for large data transfers. For smaller data transfers, and non-TCP applications like NFS, latency has a more direct impact on user experience - response time is directly proportional to latency. That is, *doubling latency doubles response time.* In these situations, the latency of the network in which a NIPS is deployed determines the acceptable latency of the NIPS.

Consider deploying a hypothetical NIPS with 1 millisecond one-way latency in the following scenarios:

- In internal corporate LANs, the round trip latency could be in the 200-300 microsecond range. Deploying our hypothetical NIPS would increase the maximum round trip latency to 2.3 milliseconds, an increase of just over 700 per cent. The time to copy a large group of files, for example, would increase by a factor of seven.
- In inter-campus corporate networks connected over a MAN, the latency could be in the 500-1000 microsecond range (or less). Deploying our hypothetical NIPS would increase the maximum round trip latency to 3 milliseconds, a minimum increase of 300 per cent. The time to copy a large group of files, for example, would increase by at least factor of three.
- Internet facing connections experience round-trip latency from 10-100 milliseconds. Deploying our hypothetical NIPS would increase the round trip latency by 1-10 per cent, which would have only a minor impact on the user experience.

The latency of the NIPS must therefore be evaluated in the context of the network in which it is deployed. For example, to protect networks that are accessed over the public Internet, one-way NIPS latencies in the 1-2 millisecond range would be acceptable. Whereas for NIPS deployments on MAN/WAN links, NIPS latencies of well under 1 millisecond would be essential. And as we have already mentioned, for deployments on internal networks where latencies are a few hundred microseconds, NIPS latencies of less than 300 microseconds would be more appropriate.

Network administrators have laboured long and hard to reduce latency within the corporate network to an absolute minimum. Core network devices such as switches are frequently chosen as much on their performance - packet loss and latency under all load conditions - as any other feature. Given that Network IPS devices are operating in-line, it is not surprising that they will be evaluated in a similar way.

For this reason, part of The NSS Group methodology uses very similar testing techniques to those we would normally employ when testing switches (in order to determine *packet latency*), in **addition** to measuring *application latency*. This group of tests determine the effect the IPS sensor has on the traffic passing through it under various load conditions. High packet latency will lower TCP throughput. High application latency will create a negative user experience.

Bi-directional network latency of UDP packets is measured under three test conditions: with no load, with 500 Mbps of HTTP traffic (or half the rated load of the device if this is less than 1Gbps), and while the device is under a heavy SYN flood attack (up to 10 per cent of the rated throughput of the sensor).

Spirent Avalanche and Reflector devices are also used to generate HTTP sessions through the device in order to gauge how any increases in latency will impact the user experience in terms of failed connections and increased Web response times. This "*application latency*" is measured both with no background load and while the device is under attack.

### Stability & Reliability

These tests verify the stability of the IPS device under various extreme conditions. Long-term stability is critical for an in-line IPS device, where failure can produce network outages.

In the first part of this test, we expose the external interface of the sensor to a constant stream of attacks 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 some legitimate sessions is transmitted through the sensor at a maximum rate of 90 per cent of the claimed throughput of the device for eight hours with no additional background traffic.

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

In the second part of the test we stress the protocol stack of the device under test by exposing it to malformed traffic from the ISIC test tool for eight hours. The device is expected to remain operational and capable of detecting and blocking exploits throughout the test to attain a PASS.

We scan the management interface for open ports and active services and report on known vulnerabilities. We also stress the protocol stack of the management interface of the NIPS by exposing it to malformed traffic from the ISIC test tool. 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 to attain a PASS. We also note whether the sensor detects the ISIC attacks even though targeted at the management port.

## Security Effectiveness

### Detection Accuracy & Breadth

This group of tests verifies that the NIPS will not block legitimate traffic (*Accuracy*) and is capable of detecting and blocking a wide range of common exploits (*Breadth*).

Although *breadth* is extremely important, *accuracy* is critical because a NIPS that blocks legitimate traffic will not remain in-line for long.

We have a number of trace files of normal traffic with "suspicious" content, together with several "neutered" exploits that have been rendered completely ineffective. The IPS attains a "PASS" for each test case if it does **not** raise an alert and does **not** block the traffic.

Whilst it is not possible to validate completely the entire signature set of any IPS, this test demonstrates how accurately the IPS detects and blocks a wide range of common exploits, port scans, and Denial of Service attempts.

This test is repeated twice: the first run with blocking disabled on the IPS in order to determine which attacks are detected and how accurately they are detected (*Attack Recognition Rating*); the second run with blocking enabled in order to determine which attacks are blocked successfully regardless of how they are detected or what alerts are raised (*Attack Blocking Rating*)

Following the initial test run, each vendor is provided with a list of CVE references of the attacks missed and is allowed 48 hours to produce an updated signature set. This updated signature set must be released to the general public as a standard signature/product update before the report is published - this ensures that vendors do not attempt to code signatures just for this test.

### Resistance To Evasion Techniques

These tests verify that the IPS is capable of detecting and blocking basic exploits when subjected to varying common evasion techniques. An IPS that cannot detect attacks subjected to these "script kiddie" evasion techniques is easily bypassed.

The tests consist of four parts:

- ***Baselines -*** *This establishes that the IPS is capable of detecting and blocking a number of common basic attacks (our baseline suite) in their normal state, with no evasion techniques applied.*
- ***Packet Fragmentation and Stream Segmentation -*** *The baseline HTTP attacks are repeated, running them through fragroute using 19 evasion techniques.*
- ***URL Obfuscation -*** *The baseline HTTP attacks are repeated, this time applying 9 URL obfuscation techniques made popular by the Whisker Web server vulnerability scanner.*
- ***Miscellaneous Evasion Techniques -*** *Certain baseline attacks are repeated, and are subjected to 7 protocol- or exploit-specific evasion techniques, including altering default ports, inserting spaces in FTP command lines, inserting non-text Telnet opcodes in FTP data streams, and RPC record fragging.*

For each of the evasion techniques, we note if (i) the attempted attack is blocked successfully (the primary aim of any IPS device), (ii) the attempted attack is detected and an alert raised in **any** form, and (iii) if the exploit is successfully "decoded" 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.

### Stateful Operation

If the IPS is tracking TCP session state, then it has the potential to introduce denial of service when the session table becomes full (too many connections) or if it can't keep up with the creation of new sessions (too many connections per second). As with latency and bandwidth, the number of connections supported by the IPS and its connection per second rate should be matched to the network.

For example, a fully saturated Gigabit Ethernet link can handle 22,000 5KByte transfers per second. Assuming each connection lasts 20 seconds, the IPS should be able to handle 448,000 simultaneous connections. These numbers scale proportionately for slower networks. Any IPS that doesn't offer these capabilities will impact performance of Web or e-commerce servers.

The aim of this section is to be able to determine whether the IPS is capable of monitoring stateful sessions established through the device at various traffic loads without either losing state or incorrectly inferring state.

An IPS that does not maintain TCP session state can flood the management console with false-positive alerts. Although this should not directly impact the IPS blocking function, it can make it very hard to perform forensic analysis of the attacks. In addition, if the default condition of the sensor is to block all traffic for which it does not believe there is a current connection in place, then an inability to maintain state under extreme conditions could result in the sensor blocking legitimate traffic by mistake.

In the first part of this test, we transmit a number of packets taken from capture files of valid exploits, but without first establishing a valid session with the target server. In order to receive a "PASS" in this test, no alerts should be raised for any of the actual exploits. However, each packet should be blocked if possible since it represents a "broken" or "incomplete" session.

In part two, we test whether the sensor is capable of preserving state across increasing numbers of open connections, as well as continuing to detect and block new exploits while not blocking legitimate traffic when the state tables are filled. Various numbers of TCP sessions from 10,000 to 1,000,000 (one million) are tested.

This test is run in both the out-of-box configuration and then repeated after applying any tuning recommended by the vendor (if applicable) to increase the size of the state tables.

## Usability

After quantitatively evaluating the network performance and security effectiveness of the IPS, we qualitatively evaluate the features and usability of the product.

This evaluation provides the reader with valuable insight into product features, how easy it is to install the IPS and perform common, day-to-day operations with the management console. Areas evaluated include *installation, configuration, policy editing, alert handling,* and *reporting and analysis*.

# TOP LAYER ATTACK MITIGATOR IPS 2400 V2.1

## Executive Summary

Top Layer's Attack Mitigator IPS is actually a family of ASIC-based *Network Intrusion Prevention Systems* (NIPS), with blocking and control against certain types of cyber attacks.

The product under test here is the Attack Mitigator IPS 2400, a combination of multiple Attack Mitigator IPS 1000 and load balancer units.

Overall, the performance of the IPS 2400 is very impressive, combining almost flawless detection rates at Gigabit wire speed with the lowest latency figures we have seen under normal traffic conditions. We also found the IPS 2400 to be very stable, surviving our extended reliability tests without missing a beat, and without blocking any legitimate traffic or succumbing to common evasion techniques.

Attack recognition capabilities are less impressive, and it should be recognised that the strongest feature of this particular product is not in its broad signature coverage or traditional IDS/IPS features, but in its ability to protect a network from the DOS and DDOS attacks which are becoming more and more prevalent in today's networks. This is the task for which the IPS 2400 was primarily designed, and is a task which it fulfils admirably.

In addition to attack mitigation, the product includes a number of features to help control and limit legitimate traffic, and the management interface is relatively easy to use both for management and monitoring.

## Architecture

Four products are available in the Attack Mitigator IPS product line:

- ***Attack Mitigator IPS 100 -*** *Designed for departmental deployments that are based on 10/100 Mbps networks. Eight 10/100 ports (four in, four out) are provided in a 2U rack mount unit.*
- ***Attack Mitigator IPS 1000 -*** *Designed for enterprises or data centre deployments, offering a Gigabit solution. The 1000 offers eight 10/100 ports (four in, four out) and two fibre Gigabit ports (one in, one out) in a 2U rack mount unit. Both the Attack Mitigator IPS 100 and 1000 models come with a 100Mbps port bypass capability. Port bypass provides basic passive redundancy in the event power is lost. This option allows the network security administrator to choose whether the Attack Mitigator IPS is deployed in fail-open or fail-closed mode.*
- ***Attack Mitigator IPS 2400 -*** *A 2x4 balanced solution, with four IPS 1000 devices and two load balancers. Although the IPS 1000 is designed for Gigabit networks, limitations in HTTP processing capabilities will restrict throughput in networks which are very heavily loaded with HTTP traffic. The IPS 2400 provides a true Gigabit throughput capability even when subjected to very high rates of HTTP connections.*
- ***Attack Mitigator IPS 4800 -*** *A 4x8 balanced solution, with eight IPS 1000 devices and four load balancers providing a fully resilient, high-availability system.*

Top Layer Networks' Attack Mitigator IPS product is a high-speed network security appliance that forms that portion of an intrusion prevention system dedicated to attack mitigation. In this attack mitigation role, the IPS Unit performs three main functions:

- ***Attack Detection -*** *The IPS Unit, by applying address matching, attack signatures, and traffic pattern analysis, detects many types of network attacks and provides the tools to identify user defined attack signatures and patterns.*
- ***Attack Mitigation -*** *Depending on the type of attack and on user-applied settings, the IPS Unit can block, limit, or just monitor each form of attack.*
- ***Attack Reporting -*** *The IPS Unit provides alarms, log entries, graphic outputs, and session records that can be used to analyse both the attacks and the performance of the IPS Unit.*

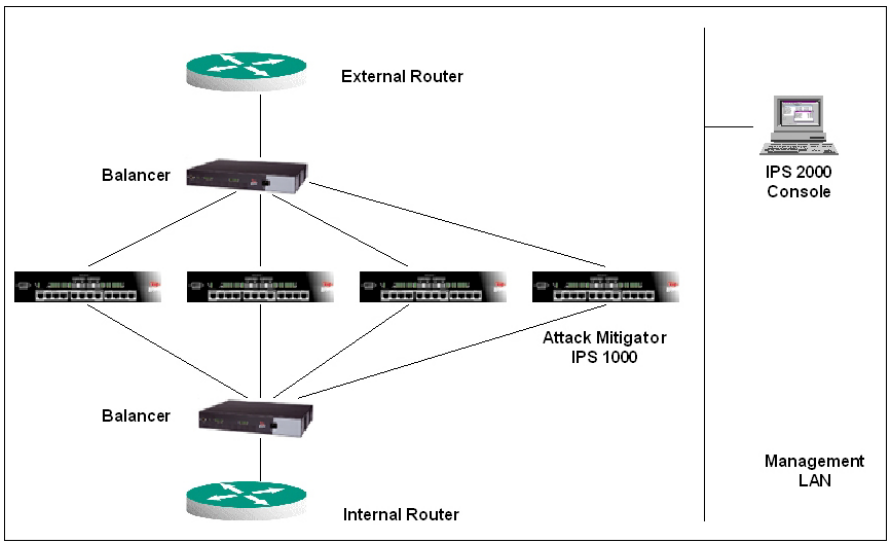


*Figure 1 - Top Layer: Attack Mitigator IPS cluster*

There is not a great deal of publicly-available information regarding the internals of the Attack Mitigator IPS, and so there is not much we can disclose other than the fact that it makes use of ASICs for performance, and runs a customised Real Time Operating System. Key software components include:

- *Packet filters*
- *Packet sequence signature evaluation tools*
- *HTTP Uniform Resource Identifier (URI) filters*
- *TCP connection counters*
- *Threat-level assessment based on observation of network incomplete connection behaviour*
- *Network application rate limits (to provide resource management by individual network applications)*

In terms of the management architecture, Top Layer has kept things as simple as possible. Where the current trend seems to be for three-tier management systems, Top Layer has eschewed that approach in favour of a two tier system with a direct relationship between management console and device being managed.

Whilst this certainly reduces complexity and potential support issues, it does not scale well, making it impossible to apply company-wide policy changes to all Top Layer devices on the network, and to consolidate alerts from multiple devices to a central point for more detailed analysis and correlation.

## Performance

The aim of this section is to verify that the sensor is capable of detecting and blocking exploits when subjected to increasing loads of background traffic up to the maximum bandwidth supported as claimed by the vendor.

For each type of background traffic, we also determine the maximum load the IPS can sustain before it begins to drop packets/miss alerts. It is worth noting that devices which demonstrate 100 per cent blocking but less than 100 per cent detection in these tests will be prone to blocking **legitimate** traffic under similar loads.

The Top Layer IPS 2400 was tested as a 1Gbps device, and performance at all levels of our load tests was impeccable, with 100 per cent of all attacks being detected and blocked under almost all conditions (the only exception to this being our 20,000 connections per second HTTP test). We would have no hesitation in rating the IPS 2400 as a true 1Gbps device.

Latency figures were outstanding at all traffic loads and with all packet sizes - under normal traffic conditions the lowest of all the devices we tested. Behaviour throughout the tests was completely predictable and very consistent, increasing only slightly as the traffic load increased.

SYN flood mitigation is the strong point of the IPS 2400, since it proxies all SYNs until it is sure that the connection is legitimate. There is obviously a performance penalty to pay for this level of protection, and latency increased significantly when the device was under constant SYN flood attack. However, at no time did the device block legitimate traffic (though delays did cause some HTTP transactions to fail) and not one invalid SYN was transmitted to the protected network.

Throughout all our tests the IPS 2400 performed consistently and reliably, continuing to pass legitimate traffic whilst blocking attack traffic when under extended attack. Likewise, exposing both the sensor and management interfaces to an extended run of ISIC-generated traffic produced had no adverse effect.

**Please refer to the *Testing Methodology* section for full details of the methodology used and performance results.**

## Security Effectiveness

We installed one Top Layer IPS 2400 sensor with the default URI filters and application library. All attack filters were set to “mitigate” mode, and no other tuning was necessary. Signature recognition (with blocking disabled) was quite poor out of the box, and the design and approach of the product made it almost impossible to improve on this significantly via the use of custom filters. In all our tests, blocking performance was significantly better than pure detection performance at 52 per cent.

As a statistic, this appears quite damning, and we would not recommend the purchase of an IPS 2400 to protect against “known” exploits of common protocols and applications. Even where the ability to exists to provide protection against such exploits - with HTTP URI filters, for example - the number of filters included out of the box is not sufficient to provide extensive coverage.

Where this product would normally be deployed is in networks which are subject to high volumes of DOS and DDOS attacks - a common threat in today’s networks. This device is capable of being installed in-line in front of a firewall and mitigating such attacks completely.

Resistance to false positives was generally very good, though in a real-world deployment the use of the “IP Unknown” blocking capability based on the default application list could give rise to a high number of false positives since it is possible that custom applications could be blocked if not defined in the ADL. We would recommend running this device in monitor mode for a few weeks to determine the optimum configuration for the application list before enabling mitigation capabilities.

The IPS 2400 performed impeccably in all of our evasion tests. In the more common evasion tests - including both *fragroute* and *Whisker* - the Top Layer device demonstrated an impeccable performance, detecting and blocking 100 per cent of all exploits (though it did not attempt to accurately decode them).

The IPS 2400 was not tricked into alerting on our stateless exploits, indicating that it would be resistant to TCP-based exploits launched via replay tools such as *Stick* and *Snot*. It was also capable of blocking these mid-flow streams completely.

Out of the box, the device maintained state on up to 500,000 open connections. It also continued to detect and block new exploits as we maintained 500,000 open connections, and no legitimate traffic was blocked during this stage of these tests. No tuning was possible to increase the number of open connections that can be supported.

Note that once the connection limit is exceeded new attacks are no longer detected (although state on existing open connections is maintained successfully until they are closed). This means that once the connection limit is exceeded, legitimate traffic may also be blocked as genuine clients attempt - and fail - to establish new connections.

**Please refer to the *Testing Methodology* section for full details of the methodology used and performance results.**

# Usability

This part of the test procedure consists of a subjective evaluation of the features and capabilities of the product, and covers *installation, configuration, policy editing, alert handling,* and *reporting and analysis*.

## Installation

In most networks, the Attack Mitigator IPS would be installed between the firewall and external router.

In this position it can provide protection for both the firewall and internal network against various DoS, DDoS or other types of attack. Positions for the IPS depend on the protection goals, and include:

- *In front of the outer firewall to protect the perimeter gateway and provide DoS and DDoS protection*
- *In front of your DMZ to protect public-facing content servers such as Web, e-mail, and FTP servers*
- *In front of internal content servers*
- *In front of interdepartmental subnets*
- *In front of a link to an extranet portion of the network*

Given the simple two-tier management structure and the appliance approach, the Attack Mitigator IPS is extremely simple to install. The clustered approach of the 2400 obviously provides additional complexities over and above those involved in installing a single device such as the IPS 1000, both in terms of cabling and management.

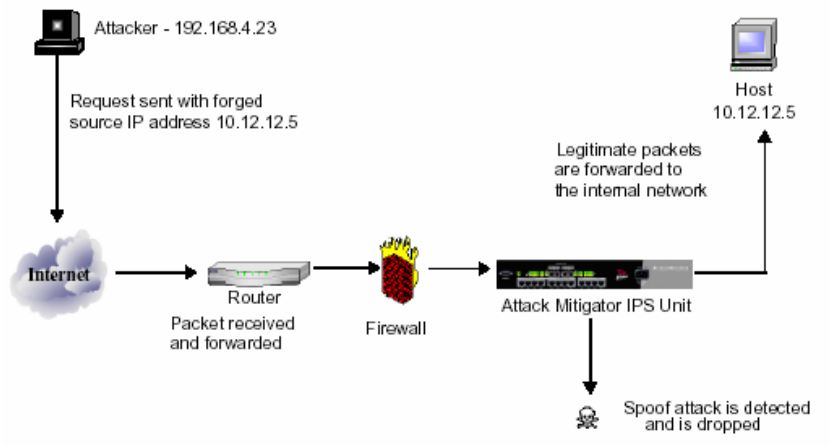


*Figure 2 - Top Layer: Typical position of an Attack Mitigator IPS on the network*

Given the fact that this is not a single-box solution, there are a number of cables to connect before the devices can be configured - eight fibre cables between the IPS 1000 units and the load balancers, six copper cables for all the management ports, and two fibre cables to provide the in-line connections. Once connected, a Perl script installed on the management console is used to provide command-line configuration and management capabilities for the load balancers (Perl is not actually provided for the Windows environment, however).

It is very important not to attempt configure the IPS 1000 devices individually, so those used to managing those devices will need to adapt to a slightly different paradigm. In a cluster configuration, the cluster itself is initialised and managed via the Perl-driven, text menu-based system which runs in a DOS window on the Windows console. Luckily, very few functions need to be carried out via that particular console, though it can prove useful when troubleshooting the cluster, providing packet distribution statistics, and so on.

One IPS 1000 is designated as a "*master*" during cluster initialisation, and from that point on, all configuration options are carried out on that device, and subsequently mirrored automatically to each of the other IPS 1000 devices in the cluster.

Unfortunately, the Web-based management interface of each individual IPS 1000 device also remains available, providing ample opportunity for the administrator to corrupt the cluster configuration completely - we would like to see the individual management capability disabled once a device becomes part of a cluster.

Having performed the initial configuration via the text-based menu system, the administrator can fire up the Web-based GUI console for all day-to-day management tasks. This is very similar to the stand-alone IPS 1000 GUI - not quite all of the IPS 1000 management features are available in the cluster management console (reports are missing, for example) but there are not too many differences, and those used to the IPS 1000 console will feel right at home.

Each Attack Mitigator IPS 1000 is a 2U rack-mount unit with two fibre Gigabit ports and twelve 10/100 Fast Ethernet ports on the front panel. The 10/100 ports are arranged in three groups of four - one group for external connections, one for internal, and the third group for management and maintenance functions. Only one interface is configured in each group initially - others can be easily configured via the management console. The two Gigabit fibre ports are also configured as one internal and one external by default.

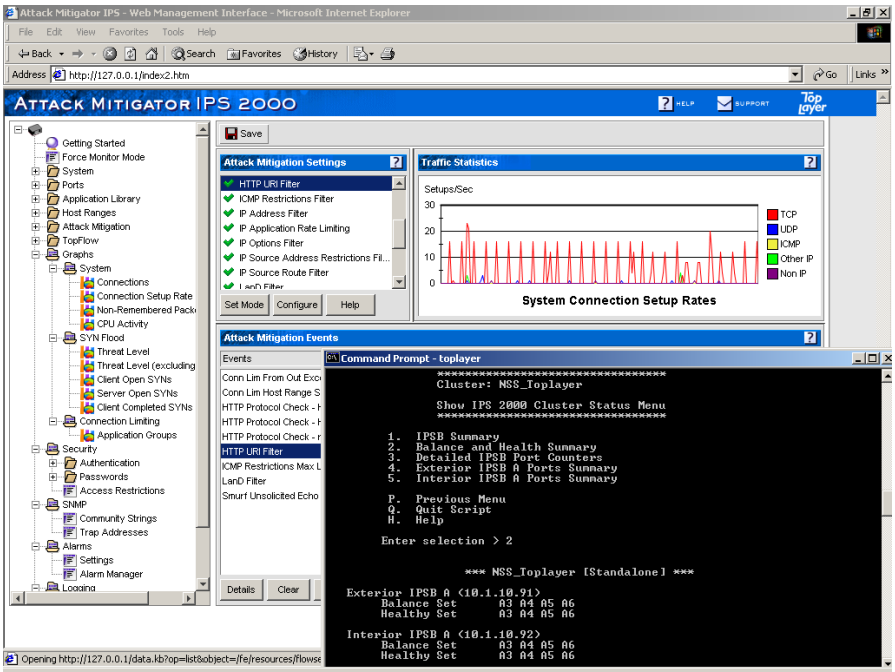


*Figure 3 - Top Layer: Managing the cluster*

Each group of ports behaves as a four port switch, and there is no way for traffic to pass directly from one switch to another without being subjected to the scrutiny of the mitigator. Nor is there any way for traffic to pass between the management ports and any of the others on the device.

In the configuration under test, only the two fibre ports were used on each device, one connected to the "*internal*" load balancer, and the other to the "*external*" load balancer. One fibre port on each load balancer is then used to provide the in-line connectivity, with each load balancer distributing traffic equally between the four sets of internal or external ports on the IPS 1000 devices.

Although this sounds cumbersome compared to some single-device IPS products, it actually works very well, and provides a high level of performance and a certain level of inbuilt resilience, given that should an individual IPS 1000 device fail, the load balancers will continue to allocate traffic - albeit at reduced performance levels - amongst the remaining devices.

Of the four maintenance ports on each IPS 1000, one is configured automatically as the port used for access to the Web-based management console. Six connections to the management network are required in total - one for each of the IPS 1000 units, and one for each of the two load balancers (although in the latest release it is no longer necessary to have a separate, private management network).

Uses for the other three maintenance ports on the device include:

- ***Forensic port -*** *Analysing various aspects of an attack (such as determining its source and the technique used) and developing a response protocol (ranging from enabling filters, notification to external agencies or service providers, and litigation or prosecution, if necessary) can be valuable components of a "defence-in-depth" strategy.*

  *Each IPS 1000 device includes a forensic port to which can be attached a network analyser or other device that records traffic. The IPS forwards **only** mitigated packets, dropped packets, or packets that are caught by the "monitor" mode to the device connected to the forensic port. In other words, only suspicious packets are directed to the forensic port.*

- ***Mirror port -*** *The IPS unit is capable of copying all of the traffic from either one internal port or one external port to a device connected to a port called a mirror port. The copied traffic includes both packets received and transmitted on that port.*

  *This capability can be useful in many applications, such as connecting a single IDS or sniffer device that is specifically tuned to inspect traffic for attacks that the IPS is not designed to detect (e-mail worms, viruses embedded in documents, and so on). Other applications include copying all traffic for statistical analysis, capacity planning, or data collection for forensic analysis. Note that **all** packets are copied to the mirror port - not just attacks - when this feature is enabled.*

As with the management ports, it is not possible to treat the other maintenance ports in isolation. If it is required to make use of the *Forensic* port, for example, it is necessary to connect the Forensic port of **each** of the four IPS 1000 devices to another switch, and then use that switch's SPAN port capability to combine that traffic for analysis by another device. Use of the *Forensic* and *Mirror* ports is somewhat cumbersome in the clustered implementation.

Although the Attack Mitigator IPS 1000 has support for *Bypass* ports, meaning that it can be configured to fail open or closed, this capability is not available in the clustered solution as tested. Failure of an individual IPS 1000 unit is handled automatically by the load balancers as already mentioned, but failure of a load balancer means the entire cluster will fail closed.

Once the cluster has been connected, powered up and initialised, it begins running immediately in “monitor” mode. This will watch traffic and evaluate it in an identical way to the “mitigate” mode, but will merely report on suspicious traffic and will not actually attempt to block it. At this point, the IPS 2400 is ready to configure in more depth.

Documentation includes a very useful *Planning and Deployment* guide and a comprehensive *Configuration and Management* guide, both of which are extensive and very thorough.

## Configuration

The Java-based *Web Management Interface* (WMI) provides access to all the configuration and management functions of the Attack Mitigator IPS 2400. As mentioned previously, configuration is actually effected on the *master* unit (this is selected automatically when the WMI is initiated), from where it is copied to all the other devices in the cluster.

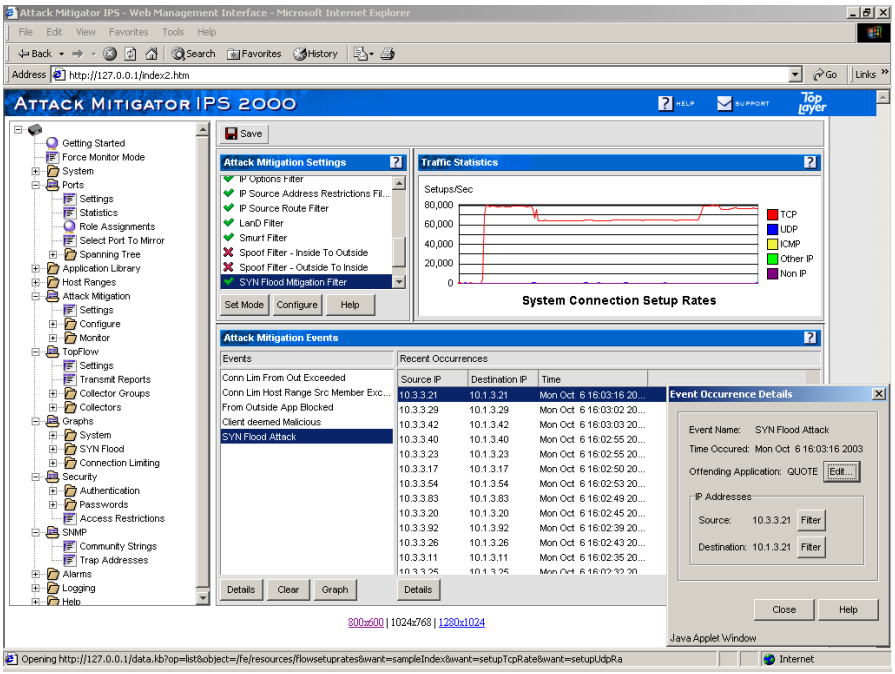


*Figure 4 - Top Layer: The Web Management Interface*

Two user accounts are provided, one for administration and one for monitoring only. It is not possible to add further user accounts, and there is no granularity of administrative functions other than these two levels. A RADIUS server can be used for authentication if required, and additional methods of access can be configured for management and configuration tasks, including Telnet, SSH, HTTPS and SNMP.

The Web-based GUI is fairly clean and uncluttered in appearance, and is reasonably intuitive in use, consisting of three panes:

- ***Banner -*** *Displayed at the top of the browser window. Provides links to access the online help table of contents, display Top Layer Support contact information, and access Top Layer’s Web site.*
- ***Navigation Tree -*** *Displayed down the left side of the browser window, containing folders that provide access to management functions.*

- ***Information and Configuration Area -*** *Displayed in the centre of the Web Management Interface window. Contains several separate windows that enable the administrator to access and configure the attack mitigation filters, examine connection set-up rates, and examine information about attack mitigation events.*

Some common tasks - such as *Attack Mitigation Settings* - are duplicated in the *Navigation Tree* and the *Information and Configuration Area*. However, the majority of tasks are available only in the *Navigation Tree*.

## Policy Management

The main building blocks for a security policy are the *Host Ranges* and *Application Library*. The *Host Ranges* folder allows the administrator to identify ranges of IP addresses that can be identified as inside and outside subnets.

The Application Library provides for identification of custom applications, and their allocation to *application groups*. It is easy to add new applications as required, and a large number are already defined in the system.

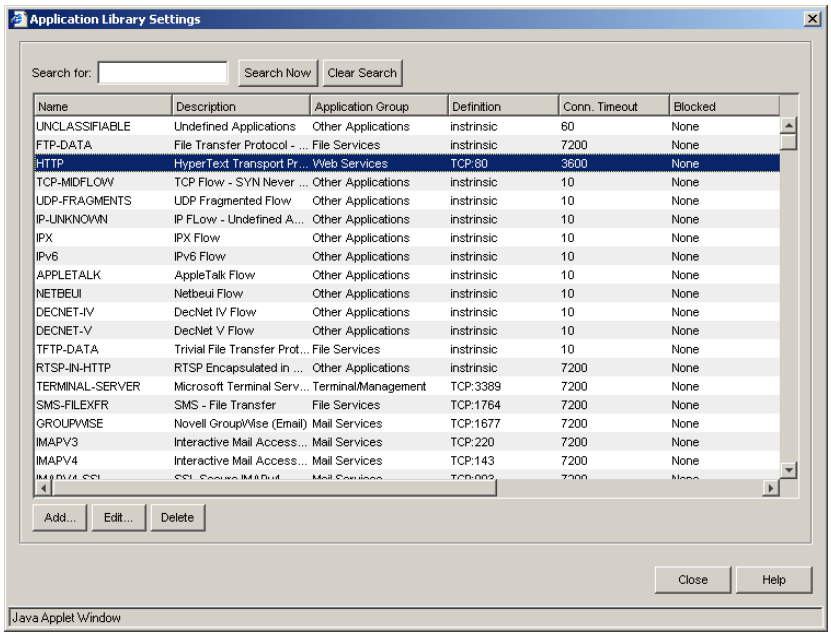


*Figure 5 - Top Layer: Application Library settings*

Applications can be defined in terms of their protocol and port number, and it is possible to block all inbound or outbound connections (or both directions) for a particular application. In this respect, the Top Layer approach appears to have more in common with the firewall model that with the IDS model.

It is also possible to set rate limiting for all custom applications (and some built-in ones) allowing the administrator to restrict inbound or outbound bandwidth on a per-application basis. In addition, the IPS provides the ability to create sets of applications called application groups and to apply connection limits to the applications in each group as a whole, and based on user defined IP address ranges. Unfortunately, it is only possible to define a maximum of eight groups, and eight are defined by default, making it necessary to delete the built-in groups before defining your own.

The IPS Unit provides a number of filters that can be configured via the *Attack Mitigation* folder:

- **Fraggle filter -** The Fraggle filter examines traffic for packets sent to the UDP ECHO port (port 7) with a destination address that would be considered a broadcast or multicast to all devices on the targeted IP network.
- **Fragment-based attacks filter -** The Fragment Restrictions filter stops incoming IP packet fragments that contain erroneous fragment offset and/or fragment length information. Handles attacks such as Boink, Teardrop and Ping of Death.
- **IP Option-based attacks filter -** This filter blocks all IP packets containing any IP option flag, including those containing source route options (which can be separately configured).
- **IP address manipulation using source route options filter -** the source route options can be used to try to circumvent firewalls or filtering routers by changing the path a packet would normally take. This filter blocks IP packets containing either the *Strict Source Route* or *Loose Source Route* option.
- **LanD filter -** A LanD attack uses an IP packet that has identical source and destination IP addresses.
- **Smurf filter -** The Smurf attack sends ICMP Echo packets (Ping) with a destination address that would be considered a broadcast or multicast to all devices on the targeted IP network. The resulting traffic can cause a flooding effect.
- **TCP/IP Fragment attacks filter -** In many network environments, IP fragmented TCP Packets do not normally exist and can be a sign of an attempt to allow an attack to evade detection (i.e. fragroute)
- **UDP Bombs filter -** This attack sends UDP packets where the length specified in the UDP header is less than the minimum legal value relative to the total packet length specified in the IP header.

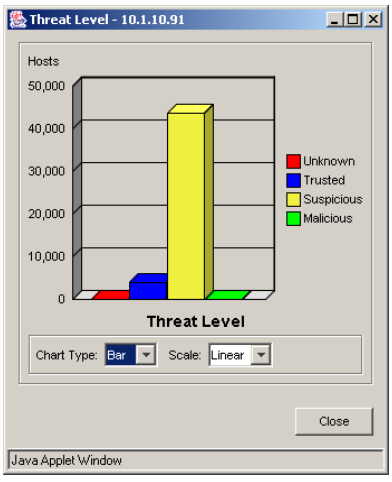


*Figure 6 - Top Layer: SYN Flood filter threat levels*

- **SYN Flood filter -** The IPS Unit provides a SYN flood monitoring mechanism that detects and responds to SYN flood attacks. When an IP address is first seen, the IPS Unit places it in either the *Unknown* or the *Suspicious* state depending on how the administrator has configured the system.

The IP address then transitions states depending on the number of open SYNs from that address exceeding one or more of three threshold values: *Trusted Threshold, Suspicious Threshold,* and *Malicious Threshold*. Configurable timeouts allow hosts to return to *Trusted* state eventually if no more attack packets are seen, and it is also possible for the administrator to reset a host to trusted state manually if required. Threat level graphs allow the administrator to display the number of hosts in each of the four threat levels (*Unknown, Trusted, Suspicious* or *Malicious*).

- **FTP Bounce filter -** By specifying an IP address that is not its own, a malicious device can cause an FTP data connection to be directed at an unsuspecting target system.
- **FTP Restricted Port filter -** In this attack, the client specifies a port that is in the assigned range (0-1023) but is not the assigned FTP data port (TCP port 20). By Spoofing the IP address of a target device (or combining this attack with the FTP Bounce attack), the attacker can cause the server to send an FTP transfer to an open port on the targeted system.

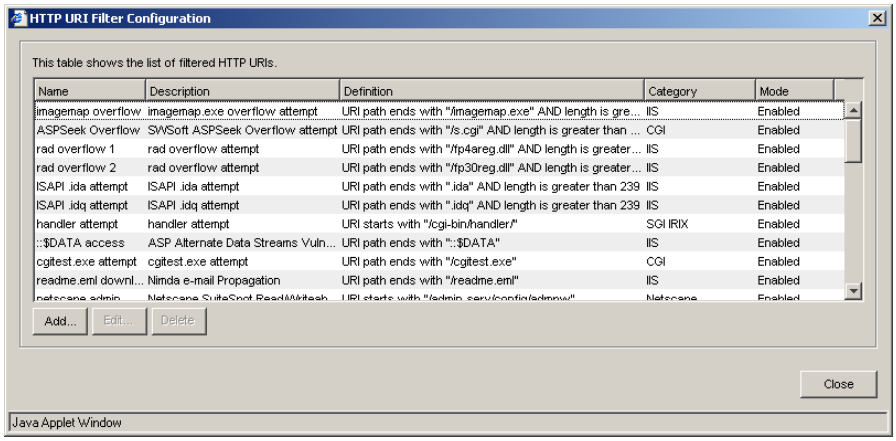


*Figure 7 - Top Layer: URI filter configuration*

- **HTTP URI filter -** The URI filter allows the administrator to check Web requests for suspicious content. The IPS performs full normalisation on the URI content (thus eliminating all attempts at evasion) including checking for attempts to pass the URI by splitting it up. If the recombined URI exceeds the 4K maximum specified for proper URIs, the IPS drops the packets and logs the event. Any number of custom URI filters can be added quickly and easily to the system, by simply specifying the length and content of the URI path. A limited number of well-tested URI filter “signatures” are also included out of the box. This is the area that provides the most scope for false positives (and thus unwitting DoS conditions if the filter is in mitigate mode) and so user-defined signature should be implemented with caution.
- **ICMP Restrictions filter -** ICMP packets that match this filter are those that exceed the user configurable *Max ICMP IP Length*, those that contain IP options, or those that have illegal header fields (non-zero unused fields or illegal Type/Code combination).
- **Application Blocking filter -** A number of network attacks (such as Trojans) masquerade as true network applications. The IPS comes with a set of predefined signatures for known malicious applications (such as Back Orifice), and when enabled, the Application Blocking filter blocks these false applications.

It is also possible to use this feature to block legitimate applications - for example, when it is necessary to block a user's access to a specific resource intensive application. For each application, the filter can be set to block traffic originating from outside the network, inside the network, or traffic from both directions. It is a simple process to add signatures for new applications to the Application Library and include them in the set of applications that the IPS device blocks. "Applications" are defined simply as protocol and port combinations, in a similar manner to a typical firewall.

- **Connection Limiting filter -** The Connection Limiting filter enables the administrator to protect network resources (such as servers and routers) from being overwhelmed by too many active connections. This feature can thus be used to protect resources from both malicious activity and non-malicious, resource-intensive traffic demands. Global connection limits can be applied to each application group and also limits, by each application group, for host ranges (which can be specified right down to an individual IP address, if required). The IPS applies connection limits to all the applications placed in a given application group. Up to eight application groups can be managed.

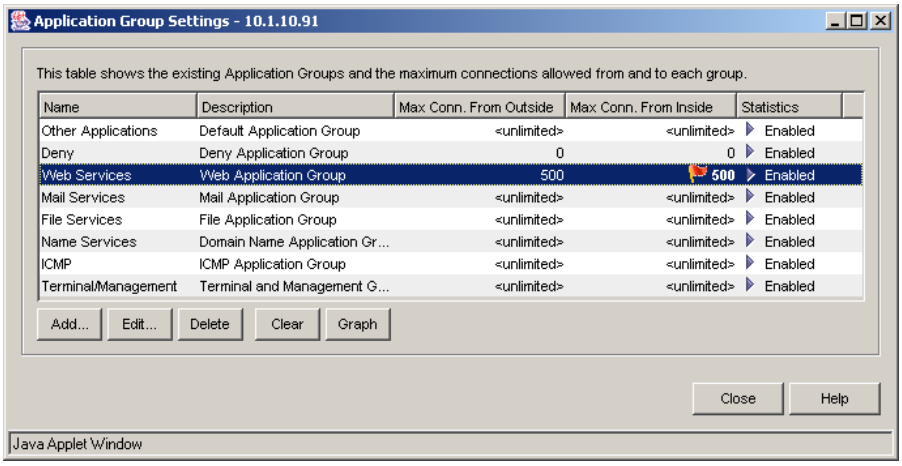


*Figure 8 - Top Layer: Limiting connections for an Application*

- **IP Application Rate Limiting filter -** This filter enables the administrator to set bandwidth limits for individual network applications. Using this filter it is possible to control connection timeout for inactive applications that do not have an explicit tear down sequence, limit traffic rates in bytes per second for traffic originating outside or inside the network, and apply URI filters to an application.
- **Outside to Inside Spoof filter -** This filter can be used to prevent attacks from being launched from outside the network where the attacker is trying to impersonate an internal device.
- **Inside to Outside Spoof filter -** This filter can be used to prevent attacks from being launched from inside the network which pretend to be from an outside entity.
- **IP Address filter** - An address filter is a designated range of IP addresses (or even a single address) whose traffic the IPS device should automatically block. This feature provides the ability to filter IP packets by source and/or destination IP address or address range.

- **IP Source Address Restriction filter -** This filter can prevent attacks where a malicious device sends a packet with a source IP address that is a broadcast, IP multicast, or IP network directed broadcast (subnet broadcast).

Via the *Attack Mitigation* folder, it is possible to set each of the attack mitigation filters to one of three modes: *Disable, Mitigate,* or *Monitor*. Each of these settings is available for every filter to provide the maximum amount of control over the attack mitigation process:

- ***Disable -*** *The IPS Unit does not watch for that type of attack or activity, and thus passes all packets, including those that include attacks. This mode does not supply an alarm, SNMP trap, or Syslog event entry.*
- ***Monitor -*** *The IPS Unit recognises the attack and supplies an alarm, SNMP trap, and Syslog event entry. However, the offending packet is forwarded to its destination. Monitor mode is useful in two situations:*
  - *To identify normal traffic patterns for a particular filter (for example, normal levels of incomplete connections)*
  - *To disable a filter but still see how the IPS Unit would handle the traffic, if the filter were enabled.*
- ***Mitigate -*** *The IPS recognises the attack and the offending packets are either dropped (but copied to the Forensic port), or limited, if the filter is a limiting filter. The IPS device generates an alarm, SNMP trap, and Syslog event.*

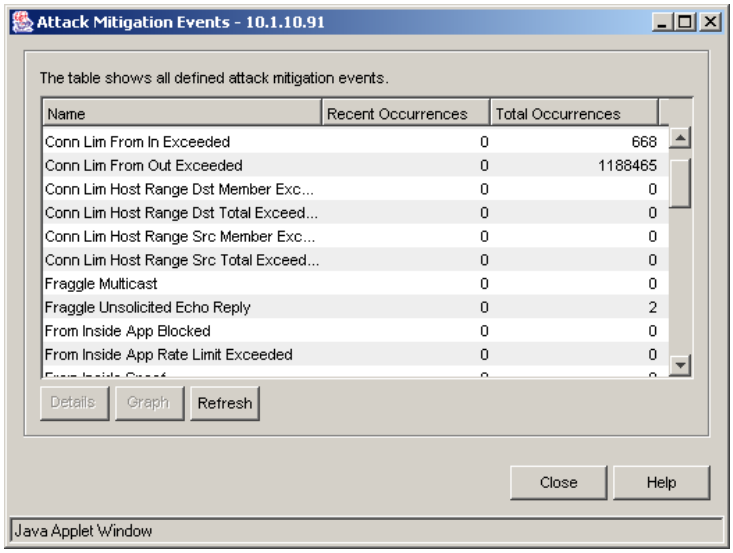


*Figure 9 - Top Layer: Viewing attack mitigation events*

In addition to setting individual filters to their Monitor settings, there is also a *Force Monitor Mode* setting. Enabling Force Monitor Mode effectively sets all enabled attack filters to *Monitor* operation (filters set to *Disable* are not affected). This is useful for temporary monitoring without mitigation, since it does not permanently affect any of the filter settings.

## Alert Handling

Where filters are in *Disabled* mode, the IPS does not monitor traffic pertaining to those filters and so no alarms are raised.

Filters in *Monitor* or *Mitigate* mode, however, will generate alarms in response to suspicious traffic which can then be logged to one or more of the following destinations:

- *Alarm Manager - stores up to 256 alarms*
- *Alarm.log files - there are six log files storing up to 1Mbyte of data each (or one day's worth) and saved in non-volatile flash memory*
- *Syslog output - if Syslog servers are configured. The latest release also provides detailed flow record reporting to external Syslog hosts*
- *SNMP Traps - if trap servers are configured*
- *TopFlow - data sent to SecureWatch collectors if configured*

The IPS Unit can suppress redundant and excessive numbers of alarms that occur during attacks. Note that if a filter is in *Mitigate* mode, in addition to raising the alarm the IPS will drop the offending packets (or limit the flow depending on the filter type) and all subsequent packets which are deemed to be part of the same flow. If a *Forensic* port is enabled, mitigated packets are copied to that port before being dropped.

A window in the *Information and Configuration Area* of the console displays events as they occur, and selecting any event brings up a list of the last ten occurrences of that event. Double clicking on an individual event brings up brief details of the attack, and it is possible from there to query the source or destination IP address in order to report on all attacks associated with a particular host.

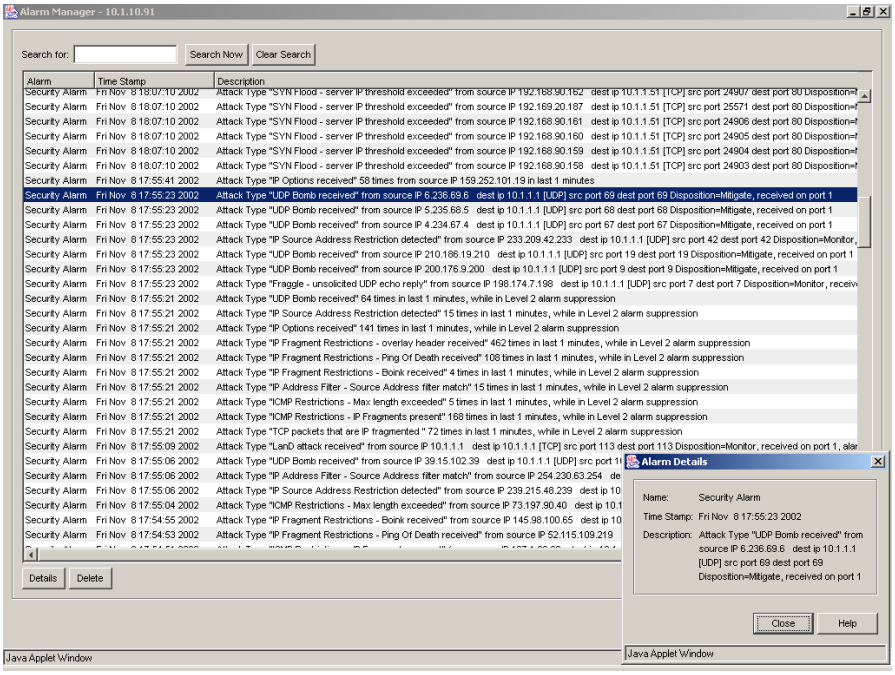


*Figure 10 - Top Layer: Alarm Manager*

The separate *Alarm Manager* window displays a list of alarms sorted by timestamps. Once again, an individual alarm entry can be selected to display brief details of the attack, and it is possible to search for particular text information in order to filter the alerts displayed - perhaps to view all "Nimda" events, for example.

The information displayed about each attack is not particularly extensive, and there is not a great deal of historical data retained, making forensic analysis difficult at anything more than the most basic level. It should be pointed out, however, that detailed forensic analysis is not the goal of this device, which is why the *Forensic* port is provided to feed attack traffic to a third party collection and analysis system if that is required.

## Reporting and Analysis

The IPS maintains an *Alarm Log* and an *Event Log*. These logs are viewable by the *Web Management Interface* (WMI) or using the CLI.

The *Alarm Log* allows the administrator to view the types of attacks that have been reported (in ASCII format text files). A new file is created each day or when the file exceeds 1 megabyte, and the IPS saves the previous five alarm logs on the CompactFlash card. The *Alarm Manager* displays a list of up to 256 alarms sorted by timestamps, and individual alarms can be selected to view the brief information on the attack.

Although the IPS 1000 can also generate security reports that can be viewed using the WMI or logged to Syslog hosts, this feature is not available in the clustered version. Instead of focussing on alert reporting and forensic analysis (which Top Layer considers to be the province of third party tools fed by the Attack Mitigator's Mirroring or Forensic ports), the IPS 2400 is designed to provide more in the way of tools to aid the administrator in monitoring and restricting bandwidth, connection rates and SYN flood or other DoS attempts.

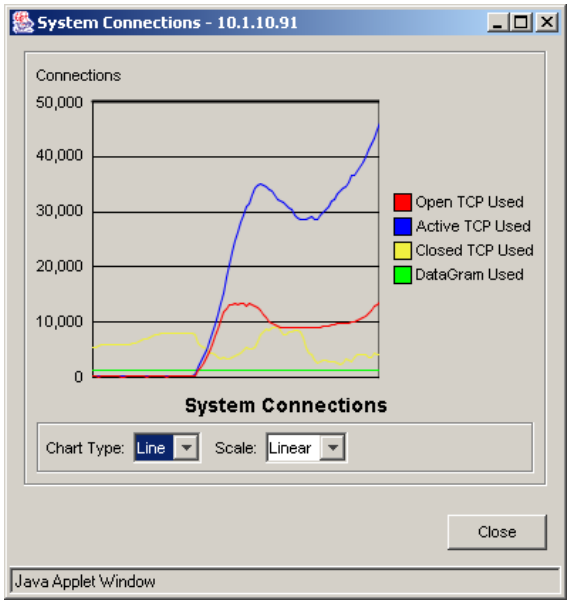


*Figure 11 - Top Layer: Connection rate graph*

The *Web Management Interface* provides connection graphs that help the administrator visualise the number of active connections for a selected application group. Two types of graphs are available:

- ***Overall Application Group Connections -*** *Indicate the number of connections initiated from both inside and outside the network for the selected application group.*

- ***Host Range Connections -*** *Indicate the number of connection initiated by an application group within a given host range, or directed to an application group within a given host range.*

These graphs can help in setting global connection limits and host range limits for each application group. The information in these graphs is updated every second and can be presented as bar graph, line graph or pie chart.

It is also possible to set the *Connection Limiting* feature to record connection statistics that can be used to better determine the proper connection limits for each host range within an application group, and for overall limits by application group.

For application groups, the IPS Unit records statistics for the following types of connections:

- ***From Outside -*** *Number of connections initiated outside the internal network for the applications in each application group*
- ***From Inside -*** *Number of connections initiated inside the internal network for the applications in each application group*

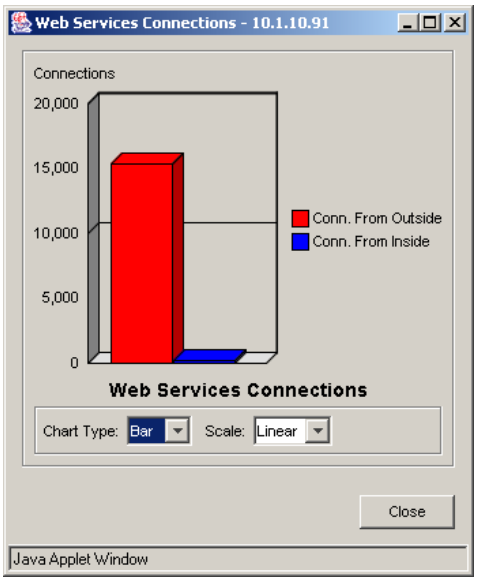


*Figure 12 - Top Layer: Connections by Application Groups*

For host ranges, statistics are recorded for the following types of connections:

- ***From Range -*** *Number of connections initiated from hosts within a given range of IP addresses (initiated by hosts outside of the internal network)*
- ***To Range -*** *Number of connections initiated to hosts within a given range of IP addresses (initiated by hosts within the internal network)*

The IPS device provides the following connection statistics for the above types of connections:

- ***Recommended Limit -*** *A calculated value based on the average plus two times the standard deviation.*

- ***Peak -*** *Largest number of concurrent, active connections seen since the IPS was booted or the counters were reset.*
- ***Average -*** *Running average number of active connections per second since the IPS was rebooted or the counters were reset.*
- ***Standard Deviation -*** *Running deviation from the average since the IPS was rebooted or the counters were reset.*

Statistics gathering can be enabled individually for each application group and for each host range within an application group. Each time an instance of statistics gathering is enabled, or connection limit settings are changed for a given application group, the previous values for recommended limit, peak, average, and standard deviation are returned to zero and the statistics gathering and calculations restart.

The IPS provides a running calculation that is updated every ten seconds, and statistics gathering can be easily enabled and disabled as required. When any of the statistics gathering instances are disabled, the values for those instances are frozen and the IPS generates an event log entry containing those statistics values. These log entries enable the administrator to keep a permanent history of the measured values.

Additional logging capabilities can be achieved using the SecureWatch traffic analysis software that is bundled with each IPS Unit. The SecureWatch collector translates and exports all events and alarms to a CheckPoint management server using the ELA protocol. The IPS Unit is OPSEC certified and can be integrated into the CheckPoint management environment.

# Verdict

### Performance

Top Layer has improved the performance of the individual IPS 1000 units considerably since we last tested them in our labs, and when clustered and load balanced in the configuration under test here the solution scales extremely well, providing excellent performance on Gigabit networks.

Performance at all levels of our load tests was impeccable, with 100 per cent of all attacks being detected and blocked under almost all conditions (the only exception to this being our 20,000 connections per second HTTP test). We would have no hesitation in rating the IPS 2400 as a true 1Gbps device.

Latency figures were outstanding at all traffic loads and with all packet sizes - the lowest that we have seen for a device of this type under normal traffic conditions. Behaviour throughout the tests was completely predictable and very consistent, increasing only slightly as the traffic load increased.

SYN flood mitigation is the strong point of the IPS 2400, since it proxies all SYNs until it is sure that the connection is legitimate. Although increased latency through the device during the SYN Flood attacks may cause problems with legitimate traffic, mitigation was complete with not one invalid SYN making it through to the protected network.

Throughout all our tests the IPS 2400 performed consistently and reliably, continuing to pass legitimate traffic whilst blocking attack traffic when under extended attack.

## Security Effectiveness

Signature recognition (with blocking disabled) was quite poor out of the box, and the design and approach of the product made it almost impossible to improve on this significantly via the use of custom filters. In all our tests, blocking performance was significantly better than pure detection performance at 52 per cent.

There is some ongoing discussion in this industry about whether a true IPS is an evolution of the firewall or the IDS. The Top Layer IPS approach is much more closely aligned to the traditional firewall model than the traditional IDS model. In fact, the Top Layer IPS 2400 has very few "traditional" IDS features, providing little in the way of built-in forensic analysis capabilities and a smaller set of "attack signatures" that you would expect to see in a typical IDS. Bear in mind also that those signatures are limited purely to HTTP URI filters, and do not cover other common protocols such as FTP or SMTP. Nor do they perform any protocol analysis.

Instead, the IPS 2400 concentrates on blocking "unknown traffic" as defined by the Application Library - very similar to the port and protocol blocking capabilities of a typical firewall - and limiting connection rates and bandwidth for specific applications and subnets

Because of this, we would not recommend the purchase of an IPS 2400 as a pure IPS/IDS product to be installed behind a firewall to protect against "known" exploits of common protocols and applications. Instead, this device would be best employed in front of a firewall where it could protect both the firewall and the network behind it from high-volume DOS and DDOS attacks - a common threat in today's networks.

Resistance to false positives was generally very good, though in a real-world deployment the use of the "IP Unknown" blocking capability based on the default application list could give rise to a high number of false positives since it is possible that custom applications could be blocked if not defined in the ADL. We would recommend running this device in monitor mode for a few weeks to determine the optimum configuration for the application list before enabling mitigation capabilities.

The IPS 2400 performed impeccably in all of our evasion tests and proved resistant to all TCP-based exploits launched via replay tools such as *Stick* and *Snot*. Out of the box, the device maintained state on up to 500,000 open connections, though it was not possible to increase this.

## Usability

Although the initial cabling up of the IPS 2400 is not as straightforward as for a single-box solution, Top Layer would normally install, connect and configure the cluster as part of its after-sales service. From the end-user's point of view, therefore, this is about as easy as it gets.

The device immediately begins operating in Monitor mode to provide an indication of which traffic would be mitigated or limited without the risk of inadvertently creating a Denial of Service condition.

It is important with in-line devices such as this that sufficient features are given over to the task of traffic profiling, and the IPS does provide some good graphical monitoring tools to help determine optimum bandwidth and connection rates for various applications before limiting traffic.

There are a number of features to help control and limit legitimate traffic, as well as mitigate malicious traffic, and the management interface is relatively easy to use both for management and monitoring. It would be nice to see some form of centralised multi-device management capabilities in a future release - at the moment, each console is limited to managing a single device at a time, and thus it is difficult to create a single policy for multiple devices and then push them out from a central point.

Alert management is extremely basic, but the main job of the Attack Mitigator IPS is to stop malicious or suspicious traffic rather than analyse it, and it does that extremely well. Most users would probably be content to leave it at the fact that the bad traffic never made it onto their network, but for those who want additional forensic analysis on the mitigated traffic, the IPS does provide the Forensic port to route that traffic to a third party collection and analysis product.

Out of the box, the filters included are biased squarely towards the more common DoS, Trojan, spoofing and flooding attacks. It is also easy enough to add URI filters to watch for malicious Web traffic if you are confident enough not to risk creating a DoS condition of your own, though the pattern-matching approach to URI filtering is rather too basic to allow for the use of complex signatures.

For those who are worried about the complexities of determining bandwidth and connection limits for legitimate resources, or who simply have no need for those capabilities, the IPS provides a good way to ensure that their network is protected from the worst, and most common, DoS and DDoS attacks.

## Contact Details

**Company name:** Top Layer Networks

**E-mail:** info@Top Layer.com

**Internet:** www.Top Layer.com

**Address:**
2400 Computer Drive
Westboro
MA 01581
USA

**Tel:** +1 508 870 1300
**Fax:** +1 508 870 9797

# APPENDIX A – TEST RESULTS

The aim of this procedure (based on V1.0 of the NSS Group Network IPS Testing Methodology) is to provide a thorough test of all the main components of an in-line Intrusion Prevention System (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 network is 100/1000Mbit Ethernet with CAT 5e cabling and a mix of Allied Telesyn AT-9816GB and AT-9812T switches (these have a mix of fibre and copper Gigabit interfaces). All IPS devices are expected to be provided as appliances - if software-only, the supplier pre-installs the software on the recommended hardware platform. The IPS is configured as a perimeter device during testing (i.e. as if installed behind the main Internet gateway/firewall). There is no firewall protecting the target subnet.

Traffic generation equipment - such as the machines generating exploits, Spirent Avalanche and Spirent Smartbits *transmit* port - is connected to the “external” network, whilst the “receiving” equipment - such as the “target” hosts for the exploits, Spirent Reflector and Smartbits *receive* port - is connected to the internal network. The IPS device under test 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 device under test, from external to internal. The same traffic is mirrored to a single SPAN port of the external gateway switch, to which an Adtech network monitoring device is connected. The Adtech AX/4000 monitors the same mirrored traffic to ensure that the total amount of traffic never exceeds 1Gbps (which would invalidate the test run).

The management interface is used to connect the IPS 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.

## Section 1 – Detection Engine

The aim of this section is to verify that the sensor is capable of detecting and blocking a wide range of common exploits accurately, whilst remaining resistant to false positives. All tests in this section are completed with **no background network load**. The latest signature pack is acquired from the vendor, and sensors are deployed with **all** available attack signatures enabled (some audit/informational signatures may be disabled).

### Test 1.1 - Attack Recognition

Whilst it is not possible to validate completely the entire signature set of any IPS sensor, this test attempts to demonstrate how accurately the sensor detects and blocks a wide range of common exploits, port scans, and Denial of Service attempts. All exploits are run with no load on the network and no IP fragmentation.

Our attack suite contains over 100 exploits covering the following areas:

- ***Test 1.1.1 -*** *Backdoors (standard ports and random ports)*
- ***Test 1.1.2 -*** *DNS*
- ***Test 1.1.3 -*** *DOS*
- ***Test 1.1.4 -*** *False negatives (common exploits which have been modified to remove or alter obvious "triggers" - this ensures that the signatures are coded for the underlying vulnerability rather than a particular exploit)*
- ***Test 1.1.5 -*** *Finger*
- ***Test 1.1.6 -*** *FTP*
- ***Test 1.1.7 -*** *HTTP*
- ***Test 1.1.8 -*** *ICMP (including unsolicited ICMP response)*
- ***Test 1.1.9 -*** *Reconnaissance*
- ***Test 1.1.10 -*** *RPC*
- ***Test 1.1.11 -*** *SSH*
- ***Test 1.1.12 -*** *Telnet*
- ***Test 1.1.13 -*** *Database*
- ***Test 1.1.14 -*** *Mail*

A wide range of vulnerable target operating systems and applications are used, and the majority of the attacks are successful, gaining root shell or administrator privileges on the target machine.

We expect all the attacks to be reported in as straightforward and clear a manner as possible (i.e. an "RDS MDAC attack" should be reported as such, rather than a "Generic IIS Attack"). Wherever possible, attacks should be identified by their assigned CVE reference. It will also be noted when a response to an exploit is considered too "noisy", generating multiple similar or identical alerts for the same attack. Finally, we will note whether the device blocks the attack packet only or the entire "suspicious" TCP session.

This test is repeated twice: the first run with blocking disabled on the sensor (monitor mode only) in order to determine which attacks are detected and how accurately they are detected (*Attack Recognition Rating*); the second run with blocking enabled in order to determine which attacks are blocked successfully regardless of how they are detected or what alerts are raised (*Attack Blocking Rating*)

The "**default**" *Attack Recognition Rating-Detect Only* (ARRD) and *Attack Recognition Rating-Block* (ARRB) are each expressed as a percentage of detected/blocked exploits against total number of exploits launched with the default signature set as received by NSS. This demonstrates how effective the sensor can be when simply deploying the default configuration.

Following the initial test run, each vendor is provided with a list of CVE references of the attacks missed, and is then allowed 48 hours to produce an updated signature set. This updated signature set **must** be released to the general public as a standard signature/product update before the report is published - this ensures that vendors do not attempt to code signatures just for this test.

The sensor is then exposed to a second round of identical tests and the "**custom**" ARRD/ARRB is determined. This demonstrates how effective the vendor is at responding to a requirement for new or updated signatures.

Both the *default* and *custom* ARRD/ARRB figures are reported.

### Test 1.2 - Resistance To False Positives

The aim of this test is to demonstrate how likely it is that a sensor raises a false positive alert - particularly critical for IPS devices.

We have a number of trace files of normal traffic with "suspicious" content, together with several "neutered" exploits which have been rendered completely ineffective. If a signature has been coded for a specific piece of exploit code rather then the underlying vulnerability, or if it relies purely on pattern matching, some of these false alarms could be alerted upon.

The IPS attains a "PASS" for each test case if it does **not** raise an alert and does **not** block the traffic. Raising an alert on any of these test cases is considered a "FAIL", since none of the "exploits" used in this test represents a genuine threat. A "FAIL" would thus indicate the chance that the IPS device could block legitimate traffic inadvertently.

- ***Test 1.2.1 -*** *False positives*

## Section 2 – IPS Evasion

The aim of this section is to verify that the sensor is capable of detecting and blocking basic exploits when subjected to varying common evasion techniques.

### Test 2.1 - Baselines

The aim of this test is to establish that the sensor is capable of detecting and blocking a number of common basic attacks (our baseline suite) in their normal state, with no evasion techniques applied.

- ***Test 2.1.1 -*** *Baseline attack replay*

### Test 2.2 - Packet Fragmentation and Stream Segmentation

The baseline HTTP attacks are repeated, running them through fragroute using various evasion techniques, including:

- ***Test 2.2.1 -*** *IP fragmentation - ordered 8 byte fragments*
- ***Test 2.2.2 -*** *IP fragmentation - ordered 24 byte fragments*
- ***Test 2.2.3 -*** *IP fragmentation - out of order 8 byte fragments*
- ***Test 2.2.4 -*** *IP fragmentation - ordered 8 byte fragments, duplicate last packet*
- ***Test 2.2.5 -*** *IP fragmentation - out of order 8 byte fragments, duplicate last packet*
- ***Test 2.2.6 -*** *IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse*
- ***Test 2.2.7 -*** *IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new)*

- ***Test 2.2.8 -*** *IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old)*
- ***Test 2.2.9 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums*
- ***Test 2.2.10 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags*
- ***Test 2.2.11 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream*
- ***Test 2.2.12 -*** *TCP segmentation - ordered 1 byte segments, duplicate last packet*
- ***Test 2.2.13 -*** *TCP segmentation - ordered 2 byte segments, segment overlap (favour new)*
- ***Test 2.2.14 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers*
- ***Test 2.2.15 -*** *TCP segmentation - out of order 1 byte segments*
- ***Test 2.2.16 -*** *TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits*
- ***Test 2.2.17 -*** *TCP segmentation - ordered 1 byte segments, segment overlap (favour new)*
- ***Test 2.2.18 -*** *TCP segmentation - out of order 1 byte segments, PAWS elimination (interleaved dup segs with older TCP timestamp options)*
- ***Test 2.2.19 -*** *IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery*

For each of the evasion techniques, we note if (i) the attempted attack is blocked successfully (the primary aim of any IPS device), (ii) the attempted attack is detected and an alert raised in **any** form, and (iii) if the exploit is successfully “decoded” 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.

## Test 2.3 - URL Obfuscation

The baseline HTTP attacks are repeated, this time applying various URL obfuscation techniques made popular by the Whisker Web server vulnerability scanner, including:

- ***Test 2.3.1 -*** *URL encoding*
- ***Test 2.3.2 -*** */./ directory insertion*
- ***Test 2.3.3 -*** *Premature URL ending*
- ***Test 2.3.4 -*** *Long URL*
- ***Test 2.3.5 -*** *Fake parameter*
- ***Test 2.3.6 -*** *TAB separation*
- ***Test 2.3.7 -*** *Case sensitivity*
- ***Test 2.3.8 -*** *Windows \ delimiter*
- ***Test 2.3.9 -*** *Session splicing*

For each of the evasion techniques, we note if (i) the attempted attack is blocked successfully, (ii) the attempted attack is detected and an alert raised in **any** form, and (iii) if the exploit is successfully “decoded” 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.

### Test 2.4 - Miscellaneous Evasion Techniques

Certain baseline attacks are repeated, and are subjected to various protocol- or exploit-specific evasion techniques, including:

- ***Test 2.4.1 -*** *Altering default ports*
- ***Test 2.4.2 -*** *Inserting spaces in FTP command lines*
- ***Test 2.4.3 -*** *Inserting non-text Telnet opcodes in FTP data stream*
- ***Test 2.4.4 -*** *Polymorphic mutation (ADMmutate)*
- ***Test 2.4.5 -*** *Altering protocol and RPC PROC numbers*
- ***Test 2.4.6 -*** *RPC record fragging*

For each of the evasion techniques, we note if (i) the attempted attack is blocked successfully, (ii) the attempted attack is detected and an alert raised in **any** form, and (iii) if the exploit is successfully "decoded" 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.

## Section 3 – Stateful Operation

The aim of this section is to be able to determine whether the IPS sensor is capable of monitoring stateful sessions established through the device at various traffic loads without either losing state or incorrectly inferring state.

### Test 3.1 - Stateless Attack Replay (Mid-Flows)

This test determines whether the sensor is resistant to stateless attack flooding tools such as *Stick* and *Snot* - these utilities are used to generate large numbers of false alerts on the protected subnet using valid source and destination addresses and a range of protocols.

The main characteristic of tools such as *Stick* and *Snot* is the fact that they generate single packets containing "trigger" patterns without first attempting to establish a connection with the target server. Whilst this can be effective in raising alerts with some stateless protocols such as UDP and ICMP, they should never be capable of raising an alert for exploits based on stateful protocols such as FTP and HTTP.

In this test, we transmit a number of packets taken from capture files of valid exploits, but without first establishing a valid session with the target server. We also remove the session tear down and acknowledgement packets so that the sensor can not "infer" that a valid connection was made.

In order to receive a "PASS" in this test, no alerts should be raised for any of the actual exploits. However, each packet should be blocked if possible since it represents a "broken" or "incomplete" session.

- ***Test 3.1.1 -*** *Stateless attack replay*

### Test 3.2 - Simultaneous Open Connections (default settings)

This test determines whether the sensor is capable of preserving state across increasing numbers of open connections, as well as continuing to detect and block new exploits when the state tables are filled.

It also attempts to determine whether or not the sensor will block legitimate traffic once state tables are filled.

A legitimate HTTP session is opened and the first packet of a two-packet exploit is transmitted. The Spirent Avalanche (on the “external” interface of the IPS sensor) then opens various numbers of TCP sessions from 10,000 to 1,000,000 (one million) with the Spirent Reflector (on the “internal” interface of the IPS sensor). The initial HTTP session is then completed with the second half of the exploit and the session is closed. If the IPS is still maintaining state on the first session established, 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 - an IPS 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.

At each step, we ensure that the IPS engine is still capable of detecting and blocking freshly-launched exploits once all the connections are open, as well as confirming that the device does not block legitimate traffic (perhaps as a result of state tables filling up). This test is run using the default sensor settings.

- ***Test 3.2.1 - Attack Detection:*** *This test ensures that the sensor continues to detect new exploits as the number of open sessions is increased in stages from 10,000 to 1,000,000*
- ***Test 3.2.2 - Attack Blocking:*** *This test ensures that the sensor continues to block new exploits as the number of open sessions is increased in stages from 10,000 to 1,000,000*
- ***Test 3.2.3 - State Preservation:*** *This test ensures that the sensor maintains the state of pre-existing sessions as the number of open sessions is increased in stages from 10,000 to 1,000,000*
- ***Test 3.2.4 - Legitimate Traffic Blocking:*** *This test ensures that the sensor does not begin to block legitimate traffic as the number of open sessions is increased in stages from 10,000 to 1,000,000*

### Test 3.3 - Simultaneous Open Connections (after tuning)

Test 3.2 is repeated after any tuning recommended by the vendor (if applicable) to increase the size of the state tables.

- ***Test 3.3.1 - Attack Detection:*** *As Test 3.2.1 following tuning*
- ***Test 3.3.2 - Attack Blocking:*** *As Test 3.2.2 following tuning*
- ***Test 3.3.3 - State Preservation:*** *As Test 3.2.3 following tuning*
- ***Test 3.3.4 - Legitimate Traffic Blocking:*** *As Test 3.2.4 following tuning*

## Section 4 – Detection/Blocking Performance Under Load

The aim of this section is to verify that the sensor is capable of detecting and blocking exploits when subjected to increasing loads of background traffic up to the maximum bandwidth supported as claimed by the vendor.

The latest signature pack is acquired from the vendor, and sensors are deployed with **all** available attack signatures enabled (some audit/informational signatures may be disabled). Each sensor is configured to **detect and block** suspicious traffic.

Our “attacker” host launches a fixed number of exploits at a target host on the subnet being protected by the IPS device. The Adtech network monitor is configured to monitor the switch SPAN port consisting of normal, exploit and background traffic, and is capable of reporting the total number of exploit packets seen on the wire as verification.

A fixed number of exploits are launched with zero background traffic to ensure the sensor is capable of detecting our baseline attacks. Once that has been established, increasing levels of varying types of background traffic are generated **through** the IPS device in order to determine the point at which the sensor begins to miss attacks - all tests are repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic (or up to the maximum rated throughput of the device should this be less than 1Gbps).

At all stages, the Adtech network monitor verifies both the overall traffic loading and the total number of exploits seen on the target subnet. An additional confirmation is provided by the target host which reports the number of exploits which actually made it through.

The *Attack Blocking Rate* (ABR) at each background load is expressed as a percentage of the number of exploits blocked by the sensor (when in blocking mode) against the number verified by the Adtech network monitor and target host. The *Attack Detection Rate* (ADR) at each background load is expressed as a percentage of the number of exploits detected by the sensor (with blocking mode disabled) against the number verified by the Adtech network monitor and target host.

For each type of background traffic, we also determine the maximum load the IPS can sustain before it begins to drop packets/miss alerts. It is worth noting that devices which demonstrate 100 per cent ABR (blocking) but less than 100 per cent ADR (detection) in these tests will be prone to blocking **legitimate** traffic under similar loads.

## Test 4.1 - UDP Traffic To Random Valid Ports

This test uses UDP packets of varying sizes generated by a **SmartBits SMB6000** with LAN-3301A 10/100/1000Mbps **TeraMetrics** cards installed. A constant stream of the appropriate mix of packets - with variable source IP addresses and ports transmitting to a single fixed IP address/port - is transmitted through the IPS device bi-directionally (maximum of 1Gbps, or 500Mbps in each direction). 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 are verified by the Adtech Gigabit network monitoring tool before each test begins. Multiple tests are run and averages taken where necessary.

This traffic does not attempt to simulate any form of “real world” network condition, and the aim of this test is purely to determine the raw packet processing capability of the IPS device, and its effectiveness at passing “useless” packets quickly in order to pass potential attack packets to the detection engine.

- ***Test 4.1.1 - 64 byte packets - maximum 1,480,000 packets per second:*** *The "torture test" - it is unlikely that any real-life network will ever see network loads of almost 1.5 million 64-byte packets per second. This test therefore measure packet processing performance under the most extreme conditions. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic.*
- ***Test 4.1.2 - 440 byte packets - maximum 260,000 packets per second:*** *This test has been included to provide a comparison with our "real world" packet mixes, since the average packet size is similar. No sessions are created during this test and there is very little for the detection engine to do in the way of protocol analysis. This test provides a reasonable indication of the ability of a device to process packets from the wire on an "average" network, and we would expect all products to demonstrate good performance levels. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic.*
- ***Test 4.1.3 - 1514 byte packets - maximum 81,720 packets per second:*** *This test is the complete opposite of the 64 byte packet test, in that we would expect every single product to be capable of returning 100 per cent detection rates across the board when using only 1514 byte packets. We have included this test 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. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic.*

## Test 4.2 - HTTP "Maximum Stress" Traffic With No Transaction Delays

HTTP is the most widely used protocol in most normal networks, as well as being one of the most widely exploited. The number of potential HTTP exploits for the protocol makes a pure HTTP network something of a torture test for the average IPS sensor.

The use of the Spirent Communications Gigabit **Avalanche** and **Reflector** allows us to create true "real world" traffic at speeds of up to 2.2 Gbps as a background load for our IPS tests. Our Avalanche configuration is capable of simulating over 2.5 million users, with over 2.5 million concurrent sessions, and almost 100,000 HTTP requests per second.

By creating genuine session-based traffic with varying session lengths, the IPS is forced to track valid 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, whilst ensuring absolute accuracy and repeatability.

The aim of this test 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.

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.

- ***Test 4.2.1 -*** *Max 2,500 new connections per second - average packet size 1200 bytes - maximum 100,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With relatively low connection rates and large packet sizes, we expect all IPS sensors to achieve 100% blocking rates throughout this test.*
- ***Test 4.2.2 -*** *Max 5,000 new connections per second - average packet size 540 bytes - maximum 230,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With average connection rates average packet sizes, this is a good approximation of a real-world production network, and we expect all IPS sensors to perform well in this test.*
- ***Test 4.2.3 -*** *Max 10,000 new connections per second - average packet size 440 bytes - maximum 280,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With average packet sizes coupled with very high connection rates, this is a strenuous test for any IPS sensor, and represents a very heavily used production network.*
- ***Test 4.2.4 -*** *Max 20,000 new connections per second - average packet size 350 bytes - maximum 360,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With small packet sizes and extremely high connection rates this is an extreme test for any IPS sensor. Not many sensors will perform well at all levels of this test.*

## Test 4.3 - HTTP “Maximum Stress” Traffic With Transaction Delays

This test is identical to Test 4.2 except that we introduce 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 utilise additional resources to track those connections.

- ***Test 4.3.1 -*** *Max 5,000 new connections per second - average packet size 540 bytes - maximum 230,000 packets per second - 10 second transaction delay - maximum 50,000 open connections. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With average connection rates average packet sizes, this is a good approximation of a real-world production network, and we expect all IPS sensors to perform well in this test.*
- ***Test 4.3.2 -*** *Max 10,000 new connections per second - average packet size 440 bytes - maximum 280,000 packets per second - 10 second transaction delay - maximum 100,000 open connections. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With average packet sizes coupled with very high connection rates, this is a strenuous test for any IPS sensor, and represents a very heavily used production network.*

## Test 4.4 - Protocol Mix Traffic

Whereas 4.2 and 4.3 provide a pure HTTP environment with varying connection rates and average packet sizes, the aim of this test is to simulate more of a “real world” environment by introducing additional protocols whilst 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, whilst less stressful than previous tests, is closer to what may be found on a heavily-utilised "normal" production network.

- ***Test 4.4.1 -*** *72% HTTP traffic (560 byte packets) + 20% FTP traffic + 6% UDP traffic (256 byte packets). Max 380 new connections per second - average packet size 555 bytes - maximum 22,000 packets per second - maximum 136 open connections. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With lower connection rates, average packets sizes and a common protocol mix, this is a good approximation of a heavily-used production network, and we expect all IPS sensors to perform well throughout this test.*

## Test 4.5 - "Real World" Traffic

This is as close as it is possible to come to a true "real world" environment under lab conditions. For this test we eliminate the Reflector device and substitute an IIS Web server installed on a dual P4 SuperMicro server with Gigabit interface. This server holds a copy of The NSS Group Web site, and is capable of handling 950Mbps of traffic. We then capture a typical client browsing session on the NSS Group Web site, accessing a mixture of menu pages, lengthy text-based reports and multiple graphical images (screen shots) and have Avalanche replay multiple identical sessions from up to **25 new users per second**.

It should be noted that whereas the goal of the previous tests is a very predictable, consistent and repeatable background load that never varies, the nature of this test means that traffic is much more "bursty" in nature.

- ***Test 4.5.1 - Pure HTTP Traffic (simulated browsing session on NSS Web site):*** *Max 100 new connections per second - 25 new users per second - average packet size 1000 bytes - maximum 110,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 950Mbps of background traffic. With genuine server responses to genuine browser sessions consisting of multiple transactions per session, this is a typical "real world" background load, albeit pure HTTP. Although the Web server and the network are extremely busy at the higher traffic loads, the "normal" connection rates and packet sizes should enable most IPS sensors to perform well at all load levels in this test.*
- ***Test 4.5.2 - Protocol Mix (72% HTTP traffic (simulated browsing sessions as 4.5.1)) + 20% FTP traffic + 6% UDP traffic (256 byte packets)):*** *Max 550 new connections per second - average packet size 900 bytes - maximum 130,000 packets per second - maximum 11,000 open connections. Repeated with 250Mbps, 500Mbps, 750Mbps and 950Mbps of background traffic. With genuine server responses to genuine browser sessions consisting of multiple transactions per session, mixed with FTP and UDP traffic, this is a typical "real world" background load. Although the Web server and the network are extremely busy at the higher traffic loads, the "normal" connection rates and packet sizes should enable most IPS sensors to perform well at all load levels in this test.*

Note that the average packet sizes during this test do not match the various studies on Internet packet size distribution (we consider the average packet size to be in the region of 440-550 bytes).

However, given that this is a test which utilises real browsing sessions against a real server on a real network, the resulting packet distribution should be considered valid.

To gauge the effects of varying (smaller) packet sizes, connection rates and transaction delays, the results of tests 4.2 - 4.4 should be examined.

# Section 5 – Latency & User Response Times

The aim of this section is to determine the effect the IPS 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.

## Test 5.1 - Latency

We use Spirent SmartFlow software and The SmartBits SMB6000 with Gigabit TeraMetrics cards to create multiple traffic flows through the IPS appliance and measure the basic throughput, packet loss, and latency through the sensor. This test - whilst 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.

SmartFlow runs through several iterations of the test varying the traffic load from 250Mbps to 1Gbps bi-directionally (500Mbps in each direction, or up to the maximum rated throughput of the device should this be less than 1Gbps) in steps of 250Mbps. This is repeated for a range of packet sizes (64 bytes, 440 bytes and 1518 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.

- ***Test 5.1.1 - Latency With No Background Traffic:*** *SmartFlow traffic is passed across the infrastructure switches and through the device (the latency of the basic infrastructure is known and is constant throughout the tests). The packet loss and average latency are recorded at each packet size and each load level from 250Mbps to 1Gbps (in 250Mbps steps).*
- ***Test 5.1.2 - Latency With Background Traffic Load:*** *The Avalanche and Reflector are configured to generate a fixed amount of background HTTP traffic through the IPS sensor (up to 50 per cent of the maximum rated bandwidth of the device under test - maximum 500Mbps - maximum 2,500 new connections per second - average packet size 540 bytes - maximum 115,000 packets per second). A 250Mbps load (125Mbps bi-directional) of SmartFlow traffic at various packet sizes (64 byte, 440 byte and 1514 byte) is then passed across the infrastructure switches and through the device and the packet loss and average latency are recorded.*
- ***Test 5.1.3 - Latency When Under Attack:*** *The Spirent WebSuite software is used to generate a fixed load of DOS/DDOS traffic of 10 per cent of the maximum rated bandwidth of the device under test (maximum 100Mbps).*

*A 250Mbps load (125Mbps bi-directional) of SmartFlow traffic at various packet sizes (64 byte, 440 byte and 1514 byte) is then passed across the infrastructure switches and through the device and the packet loss and average latency are recorded.*

### Test 5.2 - User Response Times

Avalanche and Reflector devices are used to generate HTTP sessions through the device in order to gauge how any increases in latency will impact the user experience in terms of failed connections and increased Web response times.

- ***Test 5.2.1 - Web Response With No Background Traffic:*** *The Avalanche and Reflector are configured to generate HTTP traffic through the IPS sensor (up to 50 per cent of the maximum rated bandwidth of the device under test - maximum 500Mbps - maximum 2,500 new connections per second - average packet size 540 bytes - maximum 115,000 packets per second). The minimum, maximum and average page response times and number of failed connections are recorded by Avalanche to provide an indication of the expected response times under normal traffic conditions.*
- ***Test 5.2.2 - Web Response When Under Attack:*** *The Avalanche and Reflector are configured to generate HTTP traffic through the IPS sensor as for Test 5.2.1. The Spirent WebSuite software is then used to generate DOS/DDOS traffic up to 10 per cent of the maximum rated bandwidth of the device under test (maximum 100Mbps). The minimum, maximum and average page response times and number of failed connections are recorded by Avalanche to provide an indication of the expected response times when the IPS device is under attack.*

## Section 6 – Stability & Reliability

These tests attempt to verify the stability of the device under test under various extreme conditions. Long term stability is particularly important for an in-line IPS device, where failure can produce network outages.

- ***Test 6.1.1 - Blocking Under Extended Attack:*** *For this test, we expose the external interface of the device to a constant stream of alerts 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 some 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 - 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. Results are presented as a simple PASS/FAIL. If any recognisable exploits are passed - caused by either the volume of traffic* ***or*** *the IPS device failing open for any reason - this will result in a FAIL.*

- ***Test 6.1.2 - Passing Legitimate Traffic Under Extended Attack:*** *This test is identical to 6.1.1, where we expose the external interface of the device to a constant stream of alerts over an extended period of time. The device is expected to remain operational and stable throughout this test, and to pass 100 per cent of legitimate traffic. Results are presented as a simple PASS/FAIL. If any legitimate traffic is blocked - caused by either the volume of traffic* ***or*** *the IPS device failing closed for any reason - this will result in a FAIL.*
- ***Test 6.1.3 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC:*** *This test attempts to stress the protocol stack of the device under test by exposing it to traffic from the ISIC test tool. The ISIC test tool host is connected directly to the external interface of the IPS sensor, and the ISIC target directly to the internal interface. ISIC traffic is transmitted through the IPS device (without passing through any other network equipment) and the effects noted. 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 to attain a PASS.*

# Section 7 – Management and Configuration

The aim of this section is to determine the features of the management system, together with the ability of the management port on the device under test to resist attack.

## Test 7.1 - Management Port

Clearly the ability to manage the alert data collected by the sensor is a critical part of any IDS/IPS 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.

- ***Test 7.1.1 - Open ports:*** *We will scan the open ports and active services on the management interface and report on known vulnerabilities.*

***Test 7.1.2 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC:*** *This test attempts to stress the protocol stack of the management interface of the device under test by exposing it to traffic from the ISIC test tool. The ISIC test tool host is connected directly to the management interface of the IPS sensor, and that interface is also the target. ISIC traffic is transmitted to the management interface of the IPS device (without passing through any other network equipment) and the effects noted. 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 to attain a PASS. We also note whether the ISIC attacks themselves are detected by the sensor even though targeted at the management port.*

# Top Layer Attack Mitigator IPS 2400 V2.10 Test Results

## Section 1 - Detection Engine

| Test 1.1 – Attack Recognition | Attacks | Default ARRD | Default ARRB | Custom ARRD | Custom ARRB |
|---|---|---|---|---|---|
| Test 1.1.1 - Backdoors | 5 | 0 | 5[1] | 0 | 5[1] |
| Test 1.1.2 - DNS | 2 | 0 | 0 | 0 | 0 |
| Test 1.1.3 - DOS | 12 | 5 | 8 | 5 | 8 |
| Test 1.1.4 - False negatives (modified exploits) | 13 | 5 | 8 | 5 | 8 |
| Test 1.1.5 - Finger | 4 | 0 | 0 | 0 | 0 |
| Test 1.1.6 - FTP | 5 | 1 | 1 | 1 | 1 |
| Test 1.1.7 - HTTP | 40 | 23 | 25 | 25 | 27 |
| Test 1.1.8 - ICMP | 2 | 0 | 0 | 0 | 0 |
| Test 1.1.9 - Reconnaissance | 8 | 0 | 1 | 0 | 1 |
| Test 1.1.10 - RPC | 3 | 0 | 1 | 0 | 1 |
| Test 1.1.11 - SSH | 1 | 0 | 0 | 0 | 0 |
| Test 1.1.12 - Telnet | 1 | 0 | 0 | 0 | 0 |
| Test 1.1.13 - Database | 1 | 0 | 0 | 0 | 0 |
| Test 1.1.14 - Mail | 1 | 0 | 1 | 0 | 1 |
| **Total** | **98** | **34 / 98** | **50 / 98** | **36 / 98** | **52 / 98** |

| Test 1.2 – Resistance to False Positives | Pass/Fail |
|---|---|
| Test 1.2.1 - Audiogalaxy FTP traffic | PASS |
| Test 1.2.2 - MDAC heap overflow using GET instead of POST | PASS |
| Test 1.2.3 - Retrieval of Web page containing “suspicious” URLs | PASS |
| Test 1.2.4 - Simple SMTP QUIT command | PASS |
| Test 1.2.5 - Normal NetBIOS copy of “suspicious” files | PASS |
| Test 1.2.6 - Normal NetBIOS traffic | PASS |
| Test 1.2.7 - POP3 e-mail containing “suspicious” URLs | PASS |
| Test 1.2.8 - POP3 e-mail with “suspicious” DLL attachment | PASS |
| Test 1.2.9 - POP3 e-mail with “suspicious” Web page attachment | PASS |
| Test 1.2.10 - SMTP e-mail transfer containing “suspicious” URLs | PASS |
| Test 1.2.11 - SMTP e-mail transfer with “suspicious” DLL attachment | PASS |
| Test 1.2.12 - SMTP e-mail transfer with “suspicious” Web page attachment | PASS |
| Test 1.2.13 - SNMP V3 packet with invalid request ID | PASS |
| Test 1.2.14 - Fake DNS /bin/sh buffer overflow | PASS |
| Test 1.2.15 - Inter-firewall communication traffic (looks like FW-1 RDP header firewall bypass) | PASS |
| Test 1.2.16 - Fake SQL Slammer traffic | PASS |
| Test 1.2.17 - File copy of GIF file (contains bytes which look like MS Blaster NOP sled) | FAIL[2] |
| **Total Passed** | **16 / 17** |

## Section 2 - IPS Evasion

| Test 2.1 – Evasion Baselines | Detected? | Blocked? |
|---|---|---|
| Test 2.1.1 - NSS Back Orifice ping | YES | YES |
| Test 2.1.2 - Back Orifice connection | NO | NO |
| Test 2.1.3 - FTP CWD root | NO | NO |
| Test 2.1.4 - ISAPI printer overflow | NO | NO |
| Test 2.1.5 - Showmount export lists | NO | NO |
| Test 2.1.6 - Test CGI probe (/cgi-bin/test-cgi) | YES | YES |
| Test 2.1.7 - PHF remote command execution | YES | YES |
| **Total** | **3 / 7** | **3 / 7** |

| Test 2.2 – Packet Fragmentation/Stream Segmentation | Detected? | Decoded? | Blocked? |
|---|---|---|---|
| Test 2.2.1 - IP fragmentation - ordered 8 byte fragments | YES | NO | YES |
| Test 2.2.2 - IP fragmentation - ordered 24 byte fragments | YES | NO | YES |
| Test 2.2.3 - IP fragmentation - out of order 8 byte fragments | YES | NO | YES |
| Test 2.2.4 - IP fragmentation - ordered 8 byte fragments, duplicate last packet | YES | NO | YES |

| | | | |
|---|---|---|---|
| Test 2.2.5 - IP fragmentation - out of order 8 byte fragments, duplicate last packet | YES | NO | YES |
| Test 2.2.6 - IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse | YES | NO | YES |
| Test 2.2.7 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new) | YES | NO | YES |
| Test 2.2.8 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old) | YES | NO | YES |
| Test 2.2.9 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums | YES | NO | YES |
| Test 2.2.10 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags | YES | NO | YES |
| Test 2.2.11 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream | YES | NO | YES |
| Test 2.2.12 - TCP segmentation - ordered 1 byte segments, duplicate last packet | YES | NO | YES |
| Test 2.2.13 - TCP segmentation - ordered 2 byte segments, segment overlap (favour new) | YES | NO | YES |
| Test 2.2.14 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers | YES | NO | YES |
| Test 2.2.15 - TCP segmentation - out of order 1 byte segments | YES | NO | YES |
| Test 2.2.16 - TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits | YES | NO | YES |
| Test 2.2.17 - TCP segmentation - ordered 1 byte segments, segment overlap (favour new) | YES | NO | YES |
| Test 2.2.18 - TCP segmentation - out of order 1 byte segments, PAWS elimination (interleaved dup segments with older TCP timestamp options) | YES | NO | YES |
| Test 2.2.19 - IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery | YES | NO | YES |
| **Total** | **19 / 19** | **0 / 19** | **19 / 19** |

| **Test 2.3 – URL Obfuscation** | **Detected?** | **Decoded?** | **Blocked?** |
|---|---|---|---|
| Test 2.3.1 - URL encoding | YES | YES | YES |
| Test 2.3.2 - /./ directory insertion | YES | YES | YES |
| Test 2.3.3 - Premature URL ending | YES | YES | YES |
| Test 2.3.4 - Long URL | YES | YES | YES |
| Test 2.3.5 - Fake parameter | YES | YES | YES |
| Test 2.3.6 - TAB separation | YES | YES | YES |
| Test 2.3.7 - Case sensitivity | YES | YES | YES |
| Test 2.3.8 - Windows \ delimiter | YES | YES | YES |
| Test 2.3.9 - Session splicing | YES | NO | YES |
| **Total** | **9 / 9** | **8 / 9** | **9 / 9** |

| **Test 2.4 – Miscellaneous Obfuscation Techniques** | **Detected?** | **Decoded?** | **Blocked?** |
|---|---|---|---|
| Test 2.4.1 - Altering default ports | N/A | N/A | N/A |
| Test 2.4.2 - Inserting spaces in FTP command lines | N/A | N/A | N/A |
| Test 2.4.3 - Inserting non-text Telnet opcodes in FTP data stream | N/A | N/A | N/A |
| Test 2.4.4 - Polymorphic mutation (ADMmutate) | N/A | N/A | N/A |
| Test 2.4.5 - Altering protocol and RPC PROC numbers | N/A | N/A | N/A |
| Test 2.4.6 - RPC record fragging | N/A | N/A | N/A |
| **Total** | **N/A** | **N/A** | **N/A** |

## Section 3 - Stateful Operation

| **Test 3.1 – Stateless Attack Replay** | **Alert?** | **Blocked?** | **Pass/Fail** |
|---|---|---|---|
| Test 3.1.1 - Stateless Web exploits | NO | YES[3] | PASS |
| Test 3.1.2 - Stateless FTP exploits | NO | YES[3] | PASS |

| **Test 3.2 – Simultaneous Open Connections (default settings)** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Number of open connections | 10,000 | 25,000 | 50,000 | 100,000 | 250,000 | 500,000 | 1,000,000 |
| Test 3.2.1 - Attack Detection | YES | YES | YES | YES | YES | YES | NO[4] |
| Test 3.2.2 - Attack Blocking | YES | YES | YES | YES | YES | YES | YES[4] |
| Test 3.2.3 - State Preservation | YES | YES | YES | YES | YES | YES | YES[4] |
| Test 3.2.4 - Block legitimate traffic | NO | NO | NO | NO | NO | NO | YES[4] |

| Test 3.3 – Simultaneous Open Connections (after tuning) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Number of open connections | 10,000 | 25,000 | 50,000 | 100,000 | 250,000 | 500,000 | 1,000,000 |
| Test 3.3.1 - Attack Detection | YES | YES | YES | YES | YES | YES | NO[4] |
| Test 3.3.2 - Attack Blocking | YES | YES | YES | YES | YES | YES | YES[4] |
| Test 3.3.3 - State Preservation | YES | YES | YES | YES | YES | YES | YES[4] |
| Test 3.3.4 - Block legitimate traffic | NO | NO | NO | NO | NO | NO | YES[4] |

# Section 4 - Detection/Blocking Performance Under Load

| Test 4.1 – UDP traffic to random valid ports | | 250Mbps | 500Mbps | 750Mbps | 1Gbps | Max |
|---|---|---|---|---|---|---|
| Test 4.1.1 - 64 byte packet test - max 1,480,000pps | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |
| Test 4.1.2 - 440 byte packet test - max 260,000pps | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |
| Test 4.1.3 - 1514 byte packet test - max 81,720pps | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |

| Test 4.2 – HTTP "maximum stress" traffic with no transaction delays | | 250Mbps | 500Mbps | 750Mbps | 1Gbps | Max |
|---|---|---|---|---|---|---|
| Test 4.2.1 - Max 2500 connections per second - ave packet size 1200 bytes - max 100,000 packets per second | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |
| Test 4.2.2 - Max 5000 connections per second - ave packet size 540 bytes - max 230,000 packets per second | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |
| Test 4.2.3 - Max 10000 connections per second - ave packet size 440 bytes - max 280,000 packets per second | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |
| Test 4.2.4 - Max 20000 connections per second - ave packet size 350 bytes - max 360,000 packets per second | Detected | 100% | 100% | 100% | N/A[5] | 750Mbps |
| | Blocked | 100% | 100% | 100% | N/A[5] | |

| Test 4.3 – HTTP "maximum stress" traffic with transaction delays | | 250Mbps | 500Mbps | 750Mbps | 1Gbps | Max |
|---|---|---|---|---|---|---|
| Test 4.3.1 - Max 5000 connections per second - ave packet size 540 bytes - max 230,000 packets per second - 10 sec delay - max 50,000 open connections | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |
| Test 4.3.2 - Max 10000 connections per second - ave packet size 440 bytes - max 100,000 packets per second - 10 sec delay - max 50,000 open connections | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |

| Test 4.4 – Protocol mix | | 250Mbps | 500Mbps | 750Mbps | 1Gbps | Max |
|---|---|---|---|---|---|---|
| Test 4.4.1 - 72% HTTP (540 byte packets) + 20% FTP + 4% UDP (256 byte packets). Max 380 connections per second - ave packet size 555 bytes - max 22,000 packets per second - max 136 open connections | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |

| Test 4.5 – Real World traffic | | 250Mbps | 500Mbps | 750Mbps | 1Gbps | Max |
|---|---|---|---|---|---|---|
| Test 4.5.1 - Pure HTTP (simulated browsing session on NSS Web site). Max 100 connections per second - 25 new users per second - ave packet size 1000 bytes - max 110,000 packets per second | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |
| Test 4.5.2 - Protocol mix - 72% HTTP (simulated browsing sessions as 2.5.1) + 20% FTP + 4% UDP (256 byte packets). Max 550 connections per second - ave packet size 900 bytes - max 130,000 packets per second - max 11,000 open connections | Detected | 100% | 100% | 100% | 100% | 1Gbps |
| | Blocked | 100% | 100% | 100% | 100% | |

## Section 5 - Latency & User Response Times

| Test 5.1 – Latency | Packet Size | 250Mbps | 500Mbps | 750Mbps | 1Gbps |
|---|---|---|---|---|---|
| Test 5.1.1 Average latency (µs) with no background traffic | 64 | 26.84 | 27.05 | 27.34 | 28.12 |
| | 440 | 44.49 | 44.52 | 44.59 | 44.90 |
| | 1514 | 96.35 | 96.38 | 96.44 | 96.50 |
| Test 5.1.2 Average latency (µs) with background traffic (500Mbps HTTP traffic, max 2500 connections per second - ave packet size 540 bytes - max 115,000 packets per second) | 64 | 52.48 | | | |
| | 440 | 65.66 | | | |
| | 1514 | 114.43 | | | |
| Test 5.1.3 Average latency (µs) when under attack (100Mbps SYN flood) | 64 | 181383.40 | | | |
| | 440 | 211710.60 | | | |
| | 1514 | 223349.06 | | | |

| Test 5.2 – User Response Times | Attempted Trans | Failed Trans | Min Page Response | Max Page Response | Ave Page Response |
|---|---|---|---|---|---|
| Test 5.2.1 - Web page response (ms) with no background traffic (500Mbps HTTP traffic, max 2500 connections per sec - ave packet size 540 bytes - max 115,000 packets per sec) | 1444806 | 0 | <1 | <1 | <1 |
| Test 5.2.2 - Web page response (ms) when under attack (500Mbps HTTP traffic, max 2500 connections per sec - ave packet size 540 bytes - max 115,000 packets per sec PLUS 100Mbps SYN flood) | 1449742 | 845168 | 2.942 | 16.985 | 10.633 |

## Section 6 - Stability & Reliability

| Test ID | Result |
|---|---|
| Test 6.1.1 - Blocking Under Extended Attack | PASS |
| Test 6.1.2 - Passing legitimate traffic under extended attack | PASS |
| Test 6.1.3 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC | PASS |

## Section 7 - Management Interface

| Test ID | Result |
|---|---|
| Test 7.1.1 - Open Ports | PASS[6] |
| Test 7.1.2 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC | PASS |
| Test 7.1.3 - ISIC attacks detected against management interface? | YES |

Notes:

1. Backdoors/Trojans in our tests were blocked via the “IP unknown” capabilities of the application library, and not via specific signatures. Both default and non-standard ports were handled equally well.
2. This false positive can be cleared by adding the appropriate port to the application library.
3. Requires mid-flow protection filters to be enabled
4. New connections are rejected when state tables are filled. State on existing connections is maintained when state tables are filled. No tuning is possible.
5. It was not possible to generate 20,000 cps through this device - maximum is approx 16,000cps. Thus is was not possible to run this test at 1Gbps.

### Section 1: Detection Engine

We installed one Top Layer IPS 2400 sensor with the default URI filters and application library. All attack filters were set to “mitigate” mode. No other tuning was necessary.

Signature recognition (with blocking disabled) was quite poor out of the box, and the design and approach of the product made it almost impossible to improve on this significantly via the use of custom filters.

In all our tests, blocking performance was significantly better than pure detection performance at 52 per cent.

As a statistic, this appears quite damning, and we would not recommend the purchase of an IPS 2400 to protect against “known” exploits of common protocols and applications. Even where the ability to exists to provide protection against such exploits - with HTTP URI filters, for example - the number of filters included out of the box is not sufficient to provide extensive coverage.

Where this product would normally be deployed is in networks which are subject to high volumes of DOS and DDOS attacks - a common threat in today’s networks. This device is capable of being installed in-line in front of a firewall and mitigating such attacks completely.

Resistance to false positives was generally very good, though in a real-world deployment the use of the “IP Unknown” blocking capability based on the default application list could give rise to a high number of false positives since it is possible that custom applications could be blocked if not defined in the ADL. We would recommend running this device in monitor mode for a few weeks to determine the optimum configuration for the application list before enabling mitigation capabilities.

**Section 2: IPS Evasion**

The IPS 2400 performed impeccably in all of our evasion tests (the baseline exploits used in our “miscellaneous evasion” tests were not detected, preventing us form running those tests).

In the more common evasion tests - including both *fragroute* and *Whisker* - the Top Layer device demonstrated an impeccable performance, detecting and blocking 100 per cent of all exploits (though it did not attempt to accurately decode them).

**Section 3: Stateful Operation**

The IPS 2400 was not tricked into alerting on our stateless exploits, indicating that it would be resistant to TCP-based exploits launched via replay tools such as *Stick* and *Snot*. It was also capable of blocking these mid-flow streams completely.

Out of the box, the device maintained state on up to 500,000 open connections, successfully detecting our half-open exploit as it was completed. It also continued to detect and block new exploits as we maintained 500,000 open connections, and no legitimate traffic was blocked during this stage of these tests. No tuning was possible to increase the number of open connections that can be supported.

Note that once the connection limit is exceeded new attacks are no longer detected (although state on existing open connections is maintained successfully until they are closed). This means that once the connection limit is exceeded, legitimate traffic may also be blocked as genuine clients attempt - and fail - to establish new connections.

**Section 4: Detection/Blocking Performance Under Load**

Performance at all levels of our load tests was impeccable, with 100 per cent of all attacks being detected and blocked under all conditions.

The only exception to this was Test 4.2.4, our 20,000 connections per second HTTP test.

The IPS 2400 appears to have a limit of around 16,000 HTTP connection per second, and thus we were only able to run this particular test up to 750Mbps.

In all other cases performance was flawless, and we would have no hesitation in rating the IPS 2400 as a true 1Gbps device.

**Section 5: Latency & User Response Times**

Latency figures were outstanding at all traffic loads and with all packet sizes - under normal traffic conditions the lowest of all the devices we tested. Behaviour throughout the tests was completely predictable and very consistent, increasing only slightly as the traffic load increased.

SYN flood mitigation is complete with the IPS 2400, since it proxies all SYNs until it is sure that the connection is legitimate. There is obviously a performance penalty to pay for this level of protection, and latency increased significantly when the device was under constant SYN flood attack. However, at no time did the device block legitimate traffic (though delays did cause some HTTP transactions to fail) and not one invalid SYN was transmitted to the protected network.

**Section 6: Stability & Reliability**

The IPS 2400 performed consistently and reliably throughout our tests. Under extended attack it continued to pass legitimate traffic whilst blocking attack traffic in a consistent manner.

Exposing the sensor interface to an extended run of ISIC-generated traffic produced had no adverse effect, and the device continued to detect and block all other exploits throughout and following the ISIC attack.

**Section 7: Management Interface**

Open ports on the management interface are restricted to SSH and HTTPS in order to provide management access. Port 7 is also currently open (it was previously used for “keep alive” packets) but will be disabled in a future release.

The extended ISIC attack against the management interface had no adverse effect. The IPS 2400 remained operational and capable of detecting and blocking attacks, as well as remaining able to communicate with the management console.