

McAfee IntruShield 4010 V3.1

Technical Evaluation

An NSS Group Report



First published January 2006 (Version 1.0)

Published by The NSS Group
Security Testing Laboratories
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>

©1991-2006 The NSS Group

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

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

1. The information in this Report is subject to change by The NSS Group without notice.
2. The information in this Report is believed by The NSS Group to be accurate and reliable, but is not guaranteed. All use of and reliance on this Report are at your sole risk. The NSS Group is not liable or responsible for any damages, losses or expenses arising from any error or omission in this Report.
3. NO WARRANTIES, EXPRESS OR IMPLIED ARE GIVEN BY THE NSS GROUP. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE DISCLAIMED AND EXCLUDED BY THE NSS GROUP. IN NO EVENT SHALL THE NSS GROUP BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES, OR FOR ANY LOSS OF PROFIT, REVENUE, DATA, COMPUTER PROGRAMS, OR OTHER ASSETS, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
4. This Report does not constitute an endorsement, recommendation or guarantee of any of the products (hardware or software) tested or the hardware and software used in testing the products. The testing does not guarantee that there are no errors or defects in the products, or that the products will meet your expectations, requirements, needs or specifications, or that they will operate without interruption.
5. This Report does not imply any endorsement, sponsorship, affiliation or verification by or with any companies mentioned in this report.
6. All trademarks, service marks, and trade names used in this Report are the trademarks, service marks, and trade names of their respective owners, and no endorsement of, sponsorship of, affiliation with, or involvement in, any of the testing, this Report or The NSS Group is implied, nor should it be inferred.

TABLE OF CONTENTS

INTRODUCTION	1
Intrusion Prevention Systems (IPS)	1
Host IPS (HIPS).....	2
Network IPS (NIPS).....	2
Rate-Based IPS (Attack Mitigator)	3
Detection Methods.....	3
Pattern Matching	4
Stateful Pattern Matching	4
Protocol Decode	5
Heuristic Analysis	7
Anomaly Analysis	7
Which Detection Method Is The Best?	7
Implementation Challenges.....	8
Requirements for effective prevention.....	9
The NSS Intrusion Prevention Group Test.....	10
Performance	11
Security Effectiveness	14
Usability	16
MCAFEE INTRUSHIELD 4010 V3.1.....	17
Executive Summary.....	17
Architecture.....	17
IntruShield Sensor	18
IntruShield Security Management System (ISM)	21
Update Server	22
Performance	22
Security Effectiveness	23
Usability	24
Installation.....	24
Configuration	25
Policy Management.....	28
Alert Handling	35
Reporting and Analysis.....	41
Verdict.....	43
Contact Details	45
APPENDIX A – TEST RESULTS.....	46
The Test Environment	46
Section 1 – Detection Engine	46
Section 2 – Evasion	48
Section 3 – Stateful Operation.....	50
Section 4 – Detection/Blocking Performance Under Load	52
Section 5 – Latency & User Response Times.....	56
Section 6 – Stability & Reliability	58
Section 7 – Management and Configuration.....	58
McAfee IntruShield 4010 V3.1 Test Results.....	60
Section 1 - Detection Engine	60
Section 2 - IPS Evasion.....	60
Section 3 - Stateful Operation	62
Section 4 - Detection/Blocking Performance Under Load.....	62
Section 5 - Latency & User Response Times	63
Section 6 - Stability & Reliability	63
Section 7 - Management Interface	63

TABLE OF FIGURES

Figure 1 - IntruShield: Sensor configuration	25
Figure 2 - IntruShield: Port configuration	26
Figure 3 - IntruShield: Network Console	27
Figure 4 - IntruShield: User-Defined Signature Editor	28
Figure 5 - IntruShield: Policy Editor.....	29
Figure 6 - IntruShield: Rule Set Editor	30
Figure 7 - IntruShield: Policy Editor.....	31
Figure 8 - IntruShield: Editing attack details.....	32
Figure 9 - IntruShield: Reconnaissance policy	33
Figure 10 - IntruShield: Managing policies.....	34
Figure 11 - IntruShield: Assigning rule sets and access to policies	35
Figure 12 - IntruShield: The Alert Viewer	36
Figure 13 - IntruShield: Consolidated Alert View	37
Figure 14 - IntruShield: Consolidated Views with drill down.....	38
Figure 15 - IntruShield: Viewing Exploit Alert Details.....	39
Figure 16 - IntruShield: Using Ethereal to view packet details.....	40
Figure 17 - IntruShield: Incident Viewer	41
Figure 18 - IntruShield: Creating report templates	42
Figure 19 - IntruShield: Top n Report	43

The NSS Group

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

With British headquarters, and 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 world-wide.

The NSS Group's Security Testing Laboratories are available to vendors and end-users for fully independent testing of networking, communications and security hardware and software.

The NSS Group also operates certification schemes for vendors and certification bodies, and currently provides evaluation and certification of a wide range of security products, including IDS/IPS appliances, firewalls, VPNs, Web Application firewalls, multi-function security appliances, cryptographic devices 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>.

The NSS Group awards are recognised world-wide as being the most desirable and essential when it comes to security products. Vendors consider the awards to be a crucial step in any security-related marketing campaign, whilst feedback from readers of the reports indicates that participation in an NSS Group test and/or one of the **NSS Approved** awards is a prerequisite for any security product in order to be considered for purchase.



Foreword

Following the huge success of the first comprehensive *Intrusion Prevention System* (IPS) test of its kind, The NSS Group is pleased to present the results of its third IPS Group Test, the largest so far, which includes a number of new products not included in the first two reports.

As with the first two Editions, 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.

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

Overall, the brand new test suite contains over **800 individual tests**, many of which are run multiple times, to provide the most thorough and complete evaluation of IPS products available anywhere today. The NSS Group has developed advanced testing methodologies for both *Rate-Based IPS* and *Content-Based IPS* products, since these devices are often very different in operation, although all products tested in this edition of the report are content-based.

It is worth pointing out that not every product submitted for testing receives an *NSS Approved* award. Standards are very high, and only those appearing in this report have received ***NSS Approved*** awards. For this latest round of testing, **twelve** vendors submitted a total of **fourteen** products for testing, and **ten** of these passed our stringent testing to receive ***NSS Approved***. It is heartening to note that this is a much-improved success ratio over the previous round.

We believe that our IPS test methodologies - which have been updated again for this test - will become the *de facto* standard for testing in-line Intrusion Prevention/Attack Mitigation 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. The latest **IPS Group Test** report can be viewed on-line at www.nss.co.uk/ips

Bob Walder

INTRODUCTION

In a 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*, with the latter being further sub-divided into *Content-Based* and *Rate-Based* (or *Attack Mitigation*) systems.

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 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 malicious traffic, in addition to raising an alert, it will discard the packet(s) 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.

Rate-Based IPS (Attack Mitigator)

Most NIPS products are basically IDS engines that operate in-line, and are thus dependent on protocol analysis or signature matching to recognise malicious content within individual packets (or across groups of packets). These can be classed as *Content-Based IPS* systems.

There is, however, a second breed of Network IPS that ignores packet content almost completely, instead monitoring for anomalies in network traffic that might characterise a flood attempt, scan attempt, and so on. These devices are capable of monitoring traffic flows in order to determine what is considered "normal", and applying various techniques to determine when that traffic deviates from normal. This is not always as simple as watching for high-volumes of a specific type of traffic in a short space of time, since they must also be capable of detecting "stealth" attacks, such as low-rate connection floods and slow port scan attempts.

Since these devices are concerned more with anomalies in traffic flow than packet contents, they are classed as *Rate-Based IPS* systems - and are also known as *Attack Mitigators*, as they are so effective against DOS and DDOS attacks.

Detection Methods

At one time, most Network IDS/IPS products based their alerts purely on pattern matching packet contents against a database of known signatures. Then came a new breed of offerings that approached the problem in a completely different way - by doing a full protocol analysis on the data stream. Others began to use heuristics or anomaly-based analysis to determine when an attempted attack had taken place.

Today, most IDS/IPS employ a mixture of these detection methods in a single product, though some will be more biased towards one method than another.

According to Cisco, there are five main methods of attack identification (source: Cisco Systems, *The Science of Intrusion Detection System Attack Identification*):

Pattern Matching

Pattern matching in its most basic form is concerned with the identification of a fixed sequence of bytes in a single packet. In addition to the tell-tale byte sequence, most IPS will also match various combinations of the source and destination IP address or network, source and destination port or service, and the protocol. It is also often possible to tune the signature further by specifying a start and end point for inspection within the packet, or a particular combination of TCP flags.

The more specific these parameters can be, the less inspection needs to be carried out against each packet on the wire. However, this approach can make it more difficult for systems to deal with protocols that do not live on well defined ports and, in particular, Trojans, and their associated traffic, which can usually be moved at will.

Although it is often quite simple to define a signature for a particular exploit, basic pattern matching can often be too specific, sometimes requiring multiple signatures to be defined for minor variations in exploits. They are also prone to false positives, since legitimate traffic can often contain the relatively small set of criteria supposedly used to determine when an attack is taking place.

This method is usually limited to inspection of a single packet and, therefore, does not apply well to the stream-based nature of network traffic such as HTTP sessions. This limitation gives rise to easily implemented evasion techniques.

Stateful Pattern Matching

Stateful pattern matching offers a slightly more sophisticated approach, since it takes the context of the established session into account, rather than basing its analysis on a single packet.

Stateful IPS products must consider arrival order of packets in a TCP stream and should handle matching patterns across packet boundaries. Thus, if the exploit string to be matched is *foobar*, and the exploit is split across two packets, with *foo* in one and *bar* in another, the simple packet matching IPS will miss the attack, since it will not be able to match the complete string. The stateful IPS, however, will maintain the session context and reassemble the traffic stream, once again making the complete string available to the detection engine.

This requires more resources than simple pattern matching, since the IPS now has to allocate large amounts of memory and processing power to track a potentially large number of open sessions for as long as possible. This approach does make IPS evasion that much more difficult, though far from impossible.

Direction of traffic is also important here, both in terms of quality of detection and performance.

Client-to-server traffic inspection is the process of applying detection mechanisms to the "request side" portion of a communication - for example, in HTTP this could be the "GET" request coming from a client.

Client-to-server traffic inspection is typically activated to protect all traffic whether internally or externally generated. As the size of the traffic in terms of byte count is relatively small, the processing load placed on the IPS will be lower.

Server-to-client traffic inspection is the process of finding an attack in the “response side” portion of a communication - for example, in HTTP the server-to-client traffic could be the web page and content returned from the server as a result of a “GET” request. Server-to-client traffic, as in this example, is often much larger than the client-to-server traffic in terms of byte count. As a result, the processing load that is placed on an IPS is greater for server-to-client traffic.

Some vendors do not implement server-to-client signatures at all. Often this is for performance reasons, but sometimes it is a design decision by those vendors who also offer HIPS products, which are often better placed to detect the types of exploits executed by malicious response traffic as opposed to request traffic. Some vendors do include server-to-client signatures, but recommend they are disabled when performance is paramount. Bi-directional detection can have a significant impact on performance in some cases - those products which can handle this situation with zero or minimal impact on performance are worth closer inspection (although this level of performance often comes with a higher price tag).

It should be noted that there are situations where disabling server-to-client signatures is reasonably safe, and - happily - these are usually the situations where the highest levels of performance are demanded. Typically, this would be where an IPS is deployed within the network perimeter, where it is unlikely that purely internal HTTP response traffic is likely to be malicious. Perimeter defences would normally be deployed with both client-to-server and server-to-client signatures enabled, but perimeter devices rarely have the same performance requirements as internal ones.

Protocol Decode

Protocol decode IPS take a radically different approach to simple pattern matching IPS products - though sometimes not quite as radically different as the marketing folks would have you believe. With this technique, the IPS detection engine performs a full protocol analysis, decoding and processing the packet contents in the same way that the target client or server application would. It also tends to be stateful.

Although this may seem like using a sledgehammer to crack a nut, it does have the advantage of highlighting anomalies in packet contents much more quickly than doing an exhaustive search of a signature database. It also has the advantage of greater flexibility in capturing attacks that would be very difficult - if not impossible - to catch using pure pattern-matching techniques, as well as new variations of old attacks. These are attacks which - although changing only slightly from variant to variant - would normally require a new signature in the database for the “traditional” IPS architecture, but which would be detected automatically by a complete protocol analysis.

One of the first things the protocol decode engine does is to apply rules defined by the appropriate RFCs to look for violations. This can help to detect certain anomalies such as binary data in an HTTP request, or a suspiciously long piece of data where it should not be - a sign of a possible buffer overflow attempt.

One simple example of how this might work concerns searching Telnet login strings for one of the many well-known login names that rootkits tend to leave behind on the system. A pattern matching system might scan *all* Telnet traffic for *all* these patterns, in which case the more patterns you add, the slower it becomes (not *always* the case, but a reasonable assumption for the purposes of this example).

In contrast, a protocol analysis system will decode the Telnet protocol and extract the login name. It can then perform an efficient search in a binary-search tree or a hash table for just the login name, which should scale much better as new signatures are added.

In theory, therefore, protocol decoding should offer more efficient processing of traffic and improved scalability as more signatures are added, compared to a pure pattern matching solution. In reality, pattern matching solutions rarely opt for a “brute force” approach (there are some extremely intelligent and efficient pattern matching mechanisms available), and so the differences are not always as marked as the marketing people would like us to believe.

Note also, that pattern matching and protocol decoding are not mutually exclusive, as some would lead you to believe. A protocol analysis IPS can only go so far with its protocol decodes before it too will be forced to perform some kind of pattern matching, albeit against a theoretically smaller subset of “signatures”.

One major downside, of course, is that if a completely new type of exploit does surface, it is likely that the developer will have to create new protocol decode code to handle it, whereas the pattern matching approach can allow the administrator to develop a custom signature much more quickly on site.

Protocol decoding does offer a number of advantages, however. It minimises the chance for false positives if the protocol is well defined and enforced (although false positives can be higher if the RFC is ambiguous), and can also be more broad and general to allow the IPS to detect minor variations of an exploit without having to implement separate signatures.

You may see this technique referred to in several different ways:

- *Protocol decode*
- *Protocol Anomaly Detection*
- *Protocol validation*

Each of these terms, if strictly applied, could use a slightly different approach to the problem. For example, we would expect a *protocol decode* engine to perform the sort of additional pattern matching and length checking mentioned above on the field contents in order to detect specific exploits or buffer overflows.

Pure *protocol validation* or *Protocol Anomaly Detection* engines, however, might go no further than decoding just enough to be able to determine if the packet follows the RFC to the letter. If not, they will raise an alert - but in allowing a packet to pass, they cannot be sure that the contents will not contain a means of exploit that just happens to conform with the RFC.

Beware the marketing hype in this particular area – no matter what architecture is used, the performance figures and detection rates in a live deployment will speak for themselves.

Heuristic Analysis

Heuristic-based signatures use some kind of algorithmic logic on which to base their alarm decisions. These algorithms are often statistical evaluations of the type of traffic being presented.

A good example of this type of signature is one that would be used to detect a port sweep. This signature looks for the presence of a threshold number of unique ports being touched on a particular machine. The signature may further restrict itself through the specification of the types of packets that it is interested in (that is, SYN packets). Additionally, there may be a requirement that all the probes must originate from a single source, and even that valid SYN ACK packets must be seen to be returned by the host being probed.

Signatures of this type will react differently on different networks, and can be a significant source of false positives if not tuned correctly, requiring some threshold manipulations to make them conform to the utilisation patterns on the network they are monitoring. This type of signature may be used to look for very complex relationships as well as the simple statistical example given.

Anomaly Analysis

The final approach is to forget about trying to identify the attacks directly, and concentrate instead on ignoring everything that is considered “normal”. This is known as “*anomaly-based*” IPS, and the basic principle is that, having identified what could be considered “normal” traffic on a network, then anything that falls outside those bounds could be considered an “intrusion” - or at the very least, something worthy of note. This is generally better suited to passive IDS rather than in-line IPS devices, given its propensity for false positives.

The primary strength of anomaly detection is its ability to recognise previously unseen attacks, since it is no longer concerned with knowing what an attack looks like - merely with knowing what does not constitute normal traffic. Its drawbacks, of course, include the necessity of training the system to separate noise from natural changes in normal network traffic (the installation of a new - perfectly legitimate - application somewhere on the network, for example).

Changes in standard operations may cause false alarms while intrusive activities that appear to be normal may cause missed detections. It is also difficult for these systems to name types of attacks, and this technology has a long way to go before it could be considered ready for “prime time”.

Which Detection Method Is The Best?

Which detection method to choose is a difficult question, and in all honesty, it is not one with which most of those evaluating these products should concern themselves.

Adequate performance to handle the traffic to which the sensor will be exposed, accuracy of alerts, low incidence of false positives, and centralised management and reporting/analysis tools are far more important than how the packets are processed.

In some instances, the lines blur between methodologies to the point where they become almost indistinguishable.

For example, most protocol decode analysis engines alert the user to the presence of protocol violations that are not directly related to any known attack but are “anomalous” (for example, length-based buffer overflow detection). Therefore, in this instance the engine has attributes of an anomaly-based system.

As we have already mentioned, most protocol analysis systems are also reduced to performing some form of pattern-matching process following the protocol decode. Likewise, even the most basic pattern-matching systems perform some form of protocol analysis - even if it is only for a limited range of protocols. In truth, almost all Network IPS systems are already adopting a hybrid architecture.

By and large, therefore, the *pattern-matching vs. protocol decode* debate is one of religion - something for the marketing departments to shout about. Why should the average user care what happens under the hood as long as the product does what it claims to do - detect and prevent intrusions?

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 conducted the first comprehensive IPS test of its kind, now updated in this Edition.

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/Attack Mitigator test methodologies (see section on *Testing Methodology* later in this report for detailed methodologies, updated for this latest test) 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/Attack Mitigator 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 800 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) or Attack Mitigation 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 453,000 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:

$$\text{throughput} = \text{window} / \text{RTT}$$

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 a range of differently-sized 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 of 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.

Naturally, Rate-Based IPS devices will not respond to the same attack traffic as Content-Based devices, and so for those the Detection Accuracy tests involve detecting and mitigating a wide range of rate-based attacks such as port scans, SYN floods, connection floods, and so on. We note which of these are mitigated completely, which are mitigated partially, and which require the use of built-in firewall capabilities.

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 (only the third is applicable to Rate-Based devices):

- **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*.

MCAFEE INTRUSHIELD 4010 V3.1

Executive Summary

Based on a mix of off-the-shelf and custom-designed processors, the McAfee IntruShield system is a high-performance appliance that offers real-time network intrusion detection and prevention against known, unknown, and Denial of Service attacks for enterprise networks.

The IntruShield system enables network attack detection and prevention at up to 2 Gbps, and is capable of operating in-line, or as a passive IDS, or both at the same time using different ports in the same appliance. The product line includes the *IntruShield Security Management (ISM)* system and a range of dedicated appliances covering bandwidth requirements from 100Mbps to 2Gbps, and deployment scenarios from SMB/Branch Office to ISP/MSP/Enterprise core.

The performance of the device submitted for testing - the IntruShield 4010 - was very impressive, combining near-perfect security effectiveness with excellent latency under all traffic loads. It also handled our demanding extended false positive, false-negative and evasion tests easily, and without blocking any legitimate traffic or succumbing to common evasion techniques

Management and control capabilities are outstanding, and despite the Java-based IMS looking to be ready for a change (this is currently in the pipeline), it still provides extremely powerful and flexible means of controlling anything from a single device to a large enterprise-wide deployment.

The admin domains and user roles make it easy to delegate the most fine-grained control across the largest organisation, whilst the rule-based policy definition makes it easy to define complex policies which can then be rolled out to a single sensor or the entire network at the click of a mouse. And once the policies have been applied, the alert handling and forensic analysis capabilities are feature-rich and very flexible.

The Virtual IPS capability was unique when the product was first launched, and now other vendors are clamouring to emulate it. This enables the administrator to apply separate policies right down to individual host level if required.

The internal firewall is also a useful feature, providing the means for an administrator to apply firewall rules in the core of the network without having to deploy a layer 3 device, and to implement McAfee's concept of the internal "Virtual Perimeter".

Finally, the SSL decryption capability is unique at the time of writing and enables IntruShield to decrypt selected SSL traffic on the fly and apply detection and protection capabilities to that traffic.

Architecture

The IntruShield offering is available as a range of dedicated, purpose-built hardware appliances, managed by a centralised, scalable management system.

The main components of the IntruShield system are as follows:

IntruShield Sensor

Six purpose-built IntruShield appliances are available:

- **IntruShield 1200** - a 100Mbps appliance designed for the SMB/Branch Office environment, with two 10/100Mbps detection ports and one response port.
- **IntruShield 1400** - a 200Mbps appliance designed for the SMB/Enterprise perimeter environment, with four 10/100Mbps detection ports and one response port.
- **IntruShield 2700** - a 600Mbps appliance designed for the Enterprise perimeter environment, with two 1Gbps and six 10/100Mbps detection ports, and three response ports.
- **IntruShield 3000** - a 1Gbps appliance designed for the Enterprise/Service Provider environment, with twelve 1Gbps detection ports, and two response ports.
- **IntruShield 4000** - a 2Gbps appliance designed for the Enterprise core environment, with four 1Gbps detection ports, and two response ports.
- **IntruShield 4010** - a 2Gbps appliance designed for the Enterprise/Service Provider environment, with twelve 1Gbps detection ports, and four response ports.

The device submitted for testing was the top-of-the line, newly released IntruShield 4010. Designed for high-bandwidth links in enterprise and service provider environments, it is equipped to support up to twelve SPAN ports or six in-line connections, or any combination of the aforementioned for up to 2Gbps of aggregated traffic.

The 4010 is a 2U rack mount unit which contains three redundant fans and can contain two hot-swap power supplies (a second redundant power supply needs to be purchased separately). The 4010 is equipped with the following ports:

- *Twelve monitoring GBIC ports, which can provide either fibre or copper connectivity. These allow monitoring of up to twelve SPAN ports, six segments in-line, or a combination (i.e. four in-line connections and four SPAN ports).*

The monitoring interfaces of the 4010 work in stealth mode, meaning they have no IP address and are not visible on the monitored segment. Note that the 4010 running in in-line mode fails closed, meaning that if the sensor fails, it will interrupt/block data flow. An optional hardware add-on - the Multi-Mode Optical Gigabit Fail Open Kit - provides fail open operation.

- *Four Response ports which, when operating in tapped mode, enable the sensor to inject response packets back through a switch or router.*
- *One 10/100 Management port, which is used for communication with the management server.*
- *Two RS-232C Console ports, used to set up and configure the sensor.*

High Availability (HA) options are well covered with the IntruShield product range, covering everything from multiple redundant components (such as fans and power supplies) to full-blown HA configurations with multiple IntruShield sensors. The 4010 supports active-active failover for High Availability (HA) configurations.

The two sensors that make up a failover pair are interconnected using two Gigabit failover interfaces, and each sensor mirrors all of the traffic on its monitoring interface(s) to the second sensor in the failover pair via the failover interfaces. Should either device fail it will fail closed, and the second device continues to function normally without losing state on any existing connections.

Virtual IPS (VIPS)

The IntruShield architecture allows for the creation of multiple *Virtual Intrusion Prevention Systems (VIPS)*, an extremely useful feature which was unique when IntruShield was launched, and which only now competitors are scrambling to emulate.

Up to 1000 Virtual IPS domains can be set up for specific departments, geographic locations or functions within an organisation, and security policies can then be set for each Virtual IPS. The VIPS functionality can be implemented in three ways:

- *By attributing Virtual Local Area Network (VLAN) tag(s) to a set of network resources*
- *By protecting a block of IP addresses utilising Classless Inter-domain Routing (CIDR) blocks*
- *By dedicating IntruShield system interfaces to protect the network resources in particular department, geography or organisational function.*

CIDR-based VIDS implementation allows granularity down to an individual host level. For example, DoS attacks can be identified and responded to with unique policies for individual hosts.

Dedicated Management Domains can be assigned at the sub-interface level. This allows for greater control of the personnel tasked with managing the security infrastructure. For example, operators can have rights assigned to one Management Domain while being restricted from having access to any other defined system Domains. This granularity in operator control increases the overall accountability and security of the installed system, ensuring that individual administrators are only permitted access to their own ports or appliances, to their own alerts and their own reports. This makes IntruShield one of the few devices suitable for a true Managed Services environment.

Internal Firewall

Adding firewall capabilities inside the network perimeter has considerable advantages when the firewall is also part of the IPS appliance. IPS devices are normally installed at Layer 2 as a “bump in the wire”, removing the need to allocate IP addresses to the interfaces and renumber networks on either side - a typical administrative overhead when deploying traditional Layer 3 firewall devices in the network core. In addition, the provision of multiple ports on a single appliance provides the means to firewall traffic between a number of different networks using a single device.

IntruShield incorporates a simple packet-filtering firewall capability, allowing it to filter and permit/deny packets based on IP address, port number and protocol, while eliminating the more costly firewall elements (both in terms of cost, and deployment effort) such as Network Address Translation (NAT) and Virtual Private Networks (VPN). This is not, therefore, intended as a replacement for the corporate firewall, but merely as a complement.

With firewall capabilities integrated into the IPS appliance, for example, the opportunity arises to impose far more granular security policies within the corporate network. It is possible to enforce different security policies for each pair of ports - and thus a different security policy for each subnet. Outbound HTTP traffic can therefore be permitted from all client machines, while being denied from HTTP servers, thus preventing the propagation of harmful Trojans should an HTTP server be infected. Likewise, the inbound SMTP traffic can be denied to all machines other than the mail server.

Nor is it necessary to rely on the IPS device purely for the prevention of malicious traffic. Instead, by simply applying ACL rules that restrict access to certain resources from specific parts of the corporate network, it can also be used to enforce generic corporate security policy.

The provision of firewall capabilities on the IPS device has one other important advantage, especially when deployed in the core of the network. Given that the analysis of packets and streams for malicious traffic is the most difficult task it has to perform, the elimination of a significant portion of that traffic based on simple firewall rules (i.e. simply deny all HTTP traffic to a network segment with no Web servers) can help to ensure that the detection engine is never stressed. This can potentially improve scalability of the overall security solution and permit much higher total levels of traffic to be handled by all the IPS devices on the network.

SSL Decryption

Protecting against malicious traffic contained within encrypted SSL streams has hitherto been impossible using in-line IPS devices. Historically the only practical method for protecting against SSL-encrypted attacks has been with the use of host IPS solutions, which can inspect the traffic coming into the host **after** it has been decrypted, or monitor the behaviour of the underlying system to mitigate an attack after it has entered the system.

IntruShield now provides the means to protect against SSL-encrypted attacks in-line by inspecting the SSL data stream, decrypted on the fly using a copy of the server private key securely stored on the sensor.

When a client initiates a connection request to the SSL server, IntruShield recognizes the SSL session request and monitors the SSL session initiation transaction between the client and the server. During the SSL session establishment phase, IntruShield uses the server's private key to decrypt and inspect the data and to determine the session keys. With these session keys, the IntruShield sensor can decrypt data packets for the life of the SSL connection. As an encrypted packet enters the sensor, IntruShield copies the encrypted packet, decrypts and then inspects the contents of the packet. The original packet is temporarily stored in a buffer in the sensor during the inspection phase.

In the case of non-attack SSL traffic, the original packet is released from the input buffer to the destination and the data from the detection engine is discarded. This approach ensures the integrity of the original packet and relieves the sensor from the overhead associated with re-encrypting the packet.

Upon detection of an attack, IntruShield can be configured to block the attack packet, allow the packet to pass while raising an alert, or allow the packet to pass without raising an alert.

If an attack is detected within the packet and the system is configured to block the attack, the original packet stored in the buffer is dropped and the sensor sends notification to the ISM to log and/or send an alert to the designated operator(s).

IntruShield can protect multiple SSL servers that use different private keys automatically. All SSL sessions are processed and tracked in separate input queues in the sensor. Overall performance is maintained via the use of custom silicon processing engines incorporated into the sensor.

Protection of the SSL Private Key is paramount, and IntruShield uses a number of mechanisms to ensure key confidentiality. Private keys are encrypted and exported from the SSL Server in PKCS #12 format and are imported into the IntruShield ISM via portable media, writable CD, floppy disk, etc. The encrypted key is imported into the IntruShield ISM and is encrypted again with the public key of the IntruShield sensor on which it will be used. When the Sensor is configured to perform SSL inspection, the ISM pushes the encrypted key to the sensor. The sensor decrypts the key with its private key and stores the resulting clear text SSL Private Key in volatile memory in the sensor.

Should someone gain unauthorised access to the IntruShield ISM the value of the SSL Key can not be determined without the possession of the Sensor Private Key that is generated and stored on the Sensor itself. If the Sensor is physically stolen, the unencrypted copy of the Private SSL Key is lost as soon as power is removed from the Sensor, or a re-boot of the Sensor is performed. The Private SSL Key is never transmitted or stored in unencrypted format and only exists in an unencrypted format in volatile RAM within the Sensor.

IntruShield Security Management System (ISM)

The *IntruShield Security Management (ISM)* system is the management solution for managing IntruShield sensor appliance deployments for large and distributed enterprise networks. ISM offers features to define, distribute, enforce, and audit security policies to protect critical servers, data centres, individual departments, distributed branches, and remote offices of a global business.

There are two versions of the ISM system:

- ***IntruShield Global Manager*** - best suited for global IDS/IPS deployments of many sensors. *IntruShield Global Manager runs on Windows with a MySQL database (MySQL is included), or Solaris 8 with an Oracle database (Oracle not included)*
- ***IntruShield Manager*** - can support distributed deployments of up to six sensors. *IntruShield Manager is supported only on Windows with an embedded MySQL database.*

The ISM server is a dedicated Windows or Sun Solaris platform running the Manager software. Depending on the server platform used and the amount of alert and logging activity on the network, the ISM can manage hundreds of sensors. Sensors use the 10/100 Management port to communicate with the ISM server over a secure, encrypted link.

The ISM software provides a Web-based user interface for configuring and managing the IntruShield system. IntruShield users connect to the ISM server from their workstations through a standard Web browser.

Update Server

New signatures and software patches are made available to customers over the Internet via the *Update Server*, which provides secure, fully automated, real-time signature updates without requiring any manual intervention.

According to a user-configured schedule or via a manual process, the ISM polls the *Update Server*, and compares the file on the Update Server with what is already available in the ISM server to determine what needs to be downloaded. Once it has received the update, the ISM then determines what signatures need to be pushed out to sensors based on the policy applied to the sensor.

For example, a policy defined for a Windows environment will receive only updated signatures that apply to that environment. The ISM compiles a specific update for each sensor and the update can then be pushed to sensors either manually, in an automated, real-time fashion or via automatic scheduled updates. One nice feature of IntruShield is that it maintains state completely during a signature update, and new signatures are even applied to subsequent packets of existing flows.

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.

Note that although McAfee rates the IntruShield 4010 as a 2Gbps device, it was tested to a maximum of 1Gbps in this round of testing. As you would expect with this amount of headroom, performance at all levels of our 1Gbps load tests was impeccable, with 100 per cent of all attacks being detected and blocked under all load conditions.

We also ran some tests up to the maximum of 2Gbps, and under normal network conditions, we would have no hesitation in rating the IntruShield 4010 as a true 2Gbps device.

Latency figures were excellent across the board with all packet sizes (even down to 64 byte packets) and all traffic loads. Latency ranged from 63µs with 250Mbps of 256 byte packets, to 127µs with 1Gbps of 1000 byte packets. Behaviour throughout the tests with no background traffic was very predictable, with minimal increases in latency as traffic levels increased from 250Mbps to 1Gbps across each packet size.

Placing the device under a half load of 500Mbps of HTTP traffic, we noted significant increases in latency, although all results remained below the "magic figure" of 300 microseconds - our limit for deploying in-line devices in the core of the network. HTTP response times were also very good, meaning IntruShield could be situated anywhere on a Gigabit network, either internally or at the perimeter.

Using the SYN proxy capability, SYN Flood mitigation was almost total, and 100Mbps of SYN flood traffic had a minimal effect on latency of normal traffic through the device. Average HTTP response increased from 204ms to 1716ms during the SYN flood, although no legitimate traffic was blocked during the attack.

The IntruShield 4010 performed consistently and completely reliably throughout our tests. Under eight hours of extended attack (comprising millions of exploits mixed with genuine traffic) it continued to block 100 per cent of attack traffic, whilst passing 100 per cent of legitimate traffic.

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

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

Security Effectiveness

We installed one sensor with the latest signature pack, and utilised the “All inclusive With Audit” policy provided out of the box - this policy has every attack signature and every audit signature (almost 1800 in total) enabled.

Signature recognition was excellent out of the box (95 per cent), and was increased to a perfect 100 per cent after the application of a signature pack update which was provided to us in under 24 hours. Blocking performance was identical throughout the tests.

We noted a minimum of “noise”, with very few test cases raising multiple alerts for a single exploit, and the accuracy of the exploit descriptions was high. Performance in our “false negative” tests was very good out of the box, and there is every indication that the majority of signatures are written for the underlying vulnerability rather than specific exploits. Specific exploit signatures are also included where appropriate, however, to provide more accurate identification for the administrator.

A major concern in deploying an IPS is the blocking of legitimate traffic. IntruShield’s resistance to false positives was excellent in our tests, and was increased to a perfect 100 per cent following the signature update.

IntruShield arrives with a number of default policies configured for different environments and with sensible PASS and BLOCK actions set for appropriate signatures.

Resistance to known evasion techniques was excellent, with IntruShield achieving a clean sweep across the board in all our evasion tests. *IP packet fragmentation, TCP stream segmentation, URL obfuscation, shell-code mutation, FTP evasion and RPC record fragmentation (ONC and MS-RPC)* all failed to trick IntruShield into ignoring valid attacks. Not only were the fragmented and obfuscated attacks blocked successfully, but all of them were decoded accurately as well.

The IntruShield 4010 demonstrated perfect performance in our stateful operation tests out of the box, handling over 1 million open connections with no tuning required. Default operation of the device is to age out “least used” connections when the state tables are full or resources are low - this behaviour is not configurable.

Stateless “exploits” are not alerted upon (this is correct behaviour in order to be resistant to *Stick* and *Snot* tools) and mid-flows are blocked by default. It is possible to configure the device to permit mid-flows if required.

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

Installation of the IntruShield Sensor and ISM software are very straightforward. We installed ISM on a Windows 2003 host, and all that was required was to insert the CD and run the set-up program. During installation, the *Java Runtime Environment* (JRE) is installed on the ISM host if it was not already installed.

On firing up the Console for the first time, it is necessary to add each Sensor to the *Resource Tree* using the *System Configuration Tool*. Each Sensor is given a unique name and shared secret to secure the initial public key exchange between ISM and Sensor.

At the Sensor itself, it is then necessary to initiate a CLI session via a PC attached to the serial console port. A few simple CLI commands are all that is required to set the Sensor name, shared secret and the IP configuration of the management port, at which point the Sensor establishes communication with the ISM, secures the connection, and makes itself available for remote management functions.

By default, the twelve Gigabit micro-GBIC interfaces (which can have either fibre or copper GBICs installed) come up in *In-Line* mode, each with the *Default In-Line IPS* policy applied, and thus the IntruShield Sensor is ready to run out of the box as an in-line IPS device with six in-line connections. The 4010 is capable of operating in one of three modes:

- ***SPAN or hub operating mode*** - the IntruShield Sensor is attached directly to the *Switch Port Analyser (SPAN)* port of a switch at a key point on the network in order to see all the traffic that is mirrored there
- ***Tap operating mode*** - Tap mode works through installation of an external wire tap (for GBIC ports) or built-in internal taps (for 10/100 Monitoring ports). An IntruShield sensor deployed in tap mode monitors or “sniffs” the packet information as it traverses the full-duplex network segment.
- ***In-line operating mode*** - In-line mode places a sensor directly in the network traffic path, inspecting all traffic at wire-speed as it passes through the assigned port pair.

Deployment modes can be mixed on a single sensor, allowing two ports to be deployed in-line, another two to passively monitor two separate network segments, and so on. For our tests, we configured one port pair in *In-line* mode and applied the *Default In-Line IPS* policy.

Extensive, and extremely useful, documentation in electronic format accompanies the IntruShield ISM and IntruShield sensors (hard copy versions are available to purchase). This documentation set consists of:

- [Quick Start Guide](#)
- [Getting Started Guide](#)
- [Sensor Installation and Configuration Guide](#)
- [ISM Installation Guide](#)
- [ISM Administrator's Guide](#)

Configuration

The IntruShield product line has been developed from the ground up with high-speed switched networks and large-scale distributed deployments in mind. The thought that has gone into this is evident both in the hardware and the management software, the latter demonstrating some extremely sophisticated features. One of those which will appeal most to administrators of larger distributed networks is the *administrative domain*.

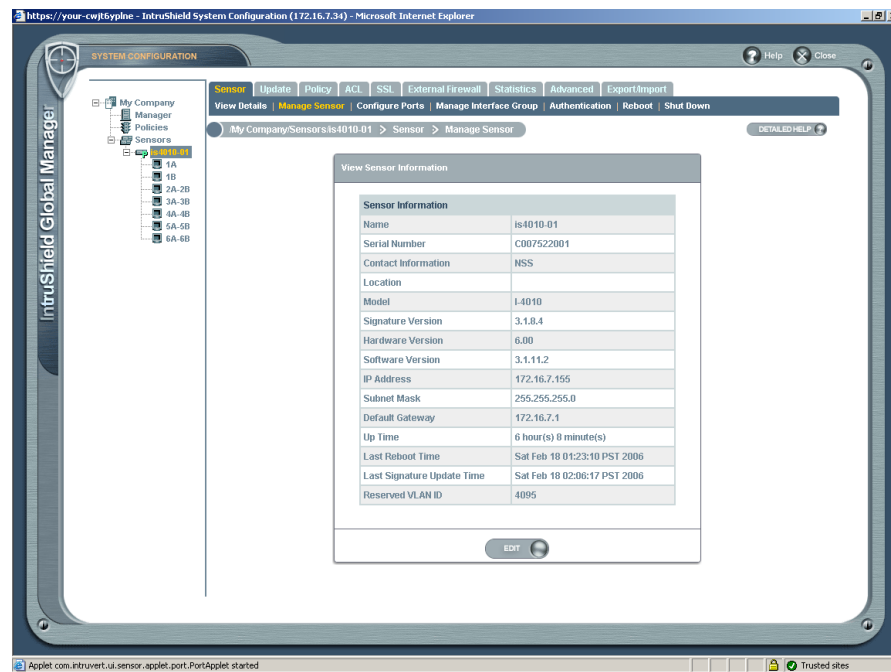


Figure 1 - IntruShield: Sensor configuration

The *admin domain* is an organisational tool used specifically to group IntruShield resources so that management of the resources can be delegated to specific IntruShield users.

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

Although it is possible to manage an entire set of IntruShield security resources from a single domain, if it is desired to delegate responsibilities for managing those resources among multiple individuals within an organisation then it is necessary to create one or more *child domains*.

The top level admin domain is called the *Root Admin Domain*, and users with *Super User* access to the Root Admin Domain have complete control over the entire administrative domain and all resources within it, including any child domains. Initially, Policies are inherited from the parent domains, but they can subsequently be changed at a child domain level.

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

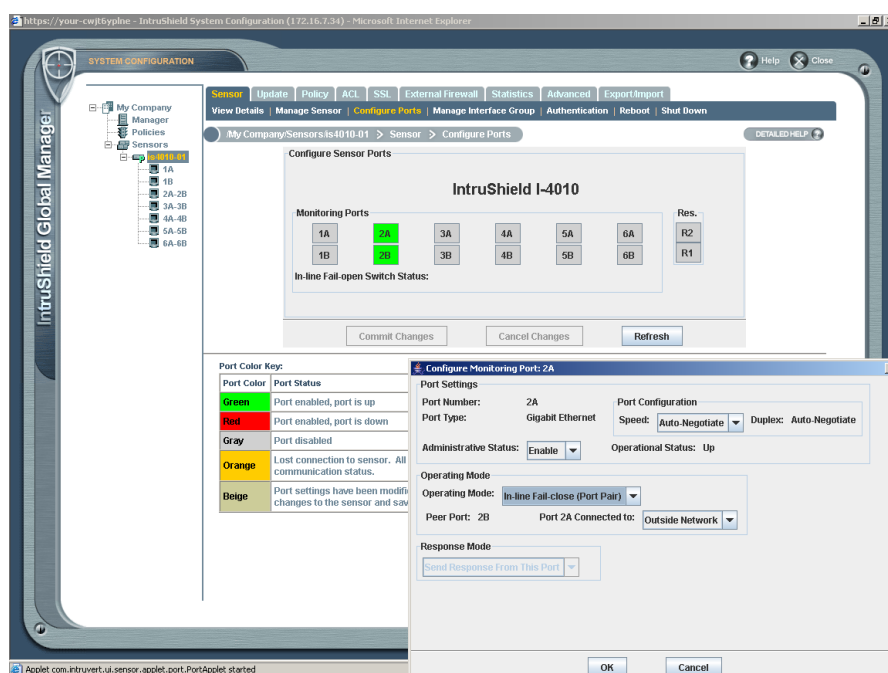


Figure 2 - IntruShield: Port configuration

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

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

The ISM Console is a Web-based Java application that can be run from any PC on the same network as the management port of the ISM server. The ISM hosts the central alert database (MySQL by default, though Oracle is also supported) and performs all the necessary alerting and reporting tasks for the IntruShield system. All the Sensors report back to the ISM host.

When first logging on to the Console, the administrator is presented with a graphical overview “dashboard” display called the *Network Console*.

This provides an at-a-glance indication of the sensor health via a coloured indicator, and an unacknowledged alert summary which shows totals of high, medium and low level alerts that have not yet been acknowledged. A hyperlink below each of these provides instant access to the *System Health Status* screen and the *Alert Viewer*, and additional buttons provide access to the *Configuration* and *Reporting* utilities.

The *System Health Status* provides a colour-coded (red, green or yellow) summary of the health of various system components, including the ISM, the Sensor and the database.

Where the status is not green, hyperlinks provide instant access to the events that caused the problem, and system faults - like alerts - can be forwarded by severity to either an SNMP or Syslog server, sent to an administrator via e-mail or pager, or processed via a script. Once the events have been investigated and resolved, they can be acknowledged and the status will return to green.

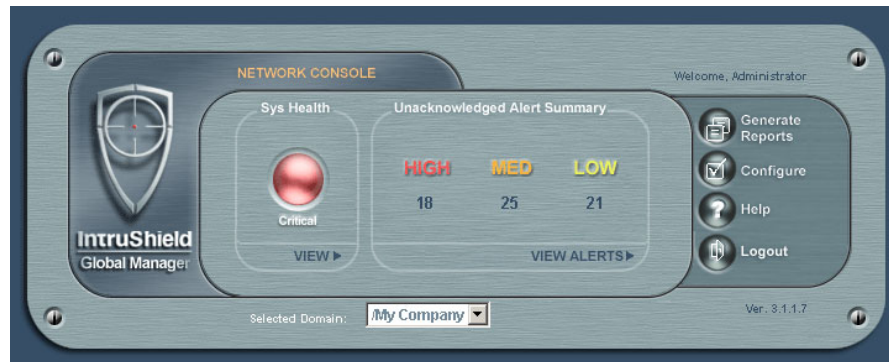


Figure 3 - IntruShield: Network Console

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

Note that signature updates can be rolled out to all sensors at the click of a button and applied to each sensor in real time without requiring a reboot. It is also possible to roll back to a previous signature pack version just as easily.

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

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

Policy Management

When working with IntruShield policies, the term *attack* is used rather than *signature*. An *attack* as defined in IntruShield is comprised of one or more *signatures*, where each signature is a method of detecting an attempt to exploit a particular vulnerability in a system.

These signatures may contain very specific means for identifying a known exploit of the vulnerability, or more generic detection methods that aid in detecting unknown exploits for the vulnerability (such as buffer overflows). Combining several such *signatures* into a single *attack* provides for maximum accuracy in attack detection.

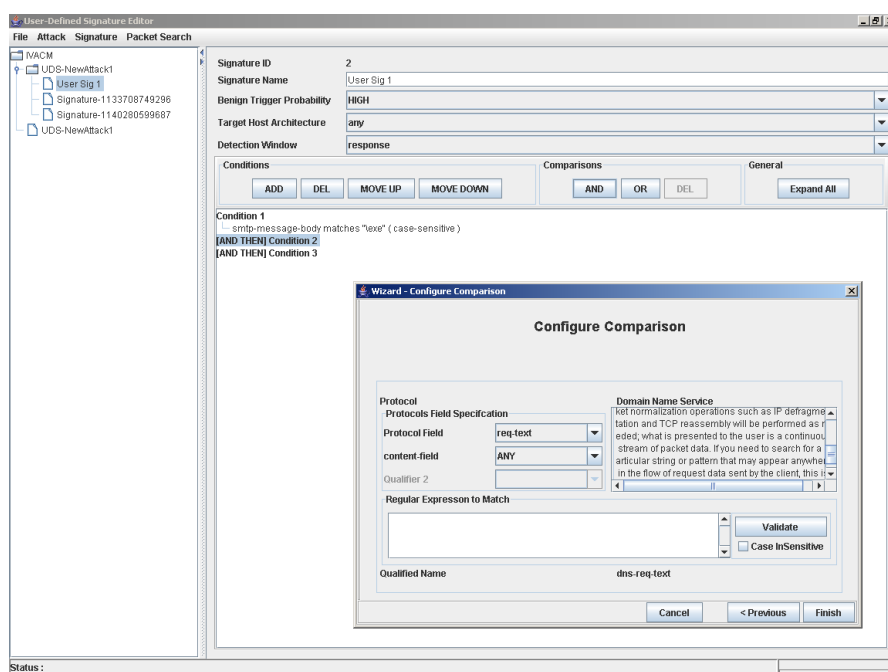


Figure 4 - IntruShield: User-Defined Signature Editor

It is possible to create user-defined signatures and attacks, and since the Wizard-driven *User Defined Signature Editor* provides access to all fields extracted by the protocol decoders as well as basic strings (such as URI, etc.), this capability is tremendously flexible and powerful.

The administrator is able to take advantage of known protocols and packages in order to perform field comparisons (i.e. check if the traffic is HTTP and the source port is 32324) or define new ones from scratch. It is also possible to perform simple packet grep string matches.

Any number of data comparisons can be added to a signature (connected with AND, OR or ANDTHEN conditions), and any number of signatures can be added to an attack. This allows for some very fine-grained control over user-defined signatures and should hopefully help to reduce false positives providing the signature is defined correctly in the first place.

McAfee supplies a set of pre-configured policies for immediate application in a number of different network environments. As well as including policies specific to particular operating systems (Windows, Solaris, etc.), server functions (Web server, mail server, FTP server, etc.) and physical locations (outside firewall, inside firewall, DMZ, etc.), there are also four *catch-all* policies: *Default IDS*, *Default In-Line IPS*, *All Inclusive With Audit* and *All Inclusive Without Audit*.

The *Default* policies include all attacks that McAfee considers are applicable to an “average” network, whilst the *All Inclusive With Audit* policy includes every single available signature - exploits and audit/informational alike. The *All Inclusive Without Audit* policy includes all available attacks, but excludes pure audit signatures.

The built-in policies are available in the *Policy Editor*, and should be considered as good starting points, designed to help get the system up and running quickly.

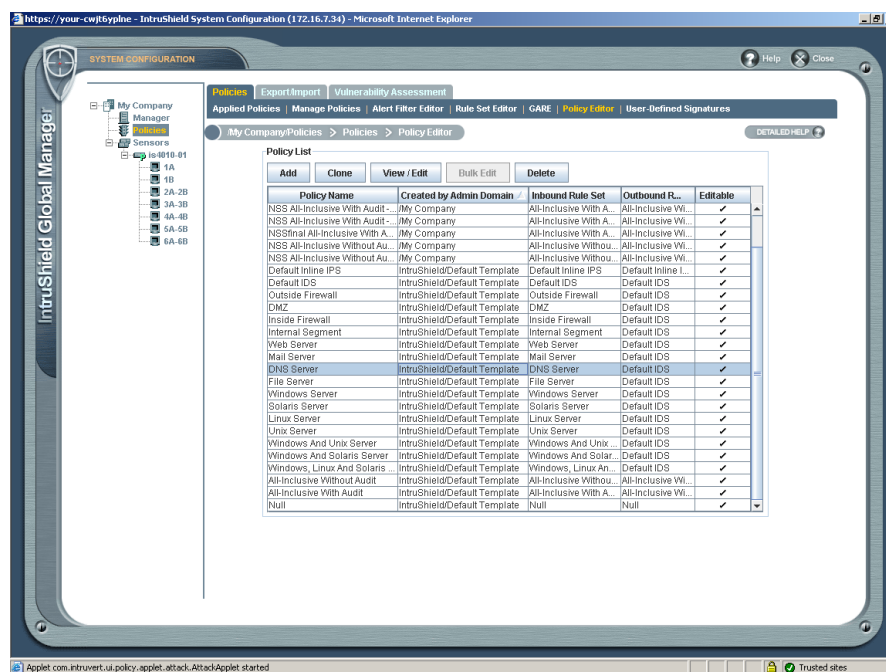


Figure 5 - IntruShield: Policy Editor

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

For many people, the *Default In-Line IPS* policy will prove more than adequate to begin with. For our testing we deployed the *All Inclusive Without Audit* policy, and the sensor performance did not appear to suffer from having all signatures enabled.

Attacks are classified into four general areas:

- **Denial of Service (DoS) and Distributed Denial of Service (DDoS):** all of the conditions indicative of activities that lead to service disruption, including the slowing down or crashing of applications, servers, or networks.
- **Exploit:** all malicious activities, other than DoS and Reconnaissance, carried out through specific traffic content. This includes viruses and worms.
- **Reconnaissance:** all of the conditions indicative of probing, scanning, and OS fingerprinting activities.

- **Policy Violation:** all activities for which the underlying traffic content may not be malicious by itself, but are explicitly forbidden by the usage policies of the administrative domain. This includes packets that violate fixed field constraints at TCP, UDP, IP, ICMP and Ethernet levels.

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

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

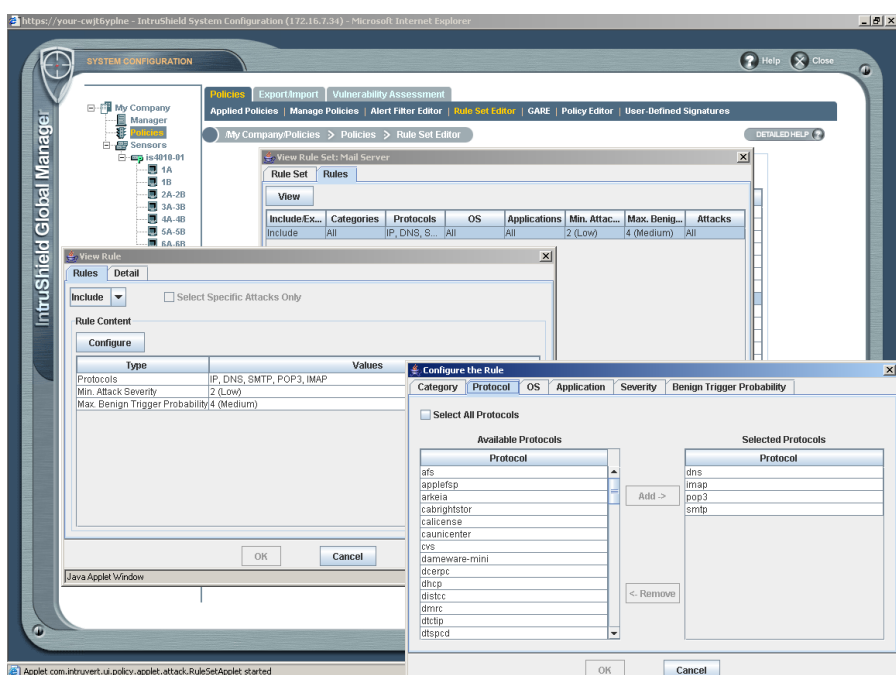


Figure 6 - IntruShield: Rule Set Editor

For example, it would be possible to include all HTTP rules but exclude the Apache-related rules if there are no Apache servers in the organisation. By this process of broadening (includes) and narrowing (excludes) the policy focus, a policy eventually comes to contain exactly the signature set that is required for a given deployment scenario.

Each attack in the IntruShield system has a wide range of informational parameters associated with it, including:

- **Attack type** (DoS, reconnaissance, exploit, etc.)
- **Severity level** (low, medium or high)
- **Benign trigger probability** (indicating the risk of false positives)
- **Protocol**
- **Target OS**
- **Target application**

The rule set can use any or all of these parameters in order to provide the most highly focussed policy.

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

That way, only the relevant signatures are applied, and the administrator can be reasonably sure that false positives will not cause a self-inflicted Denial of Service condition by ensuring that only the most focussed and accurate signatures are applied in in-line mode. A separate interface on the IntruShield could then be used to monitor the same DMZ segment in SPAN mode with a broader policy, just in case something slips through.

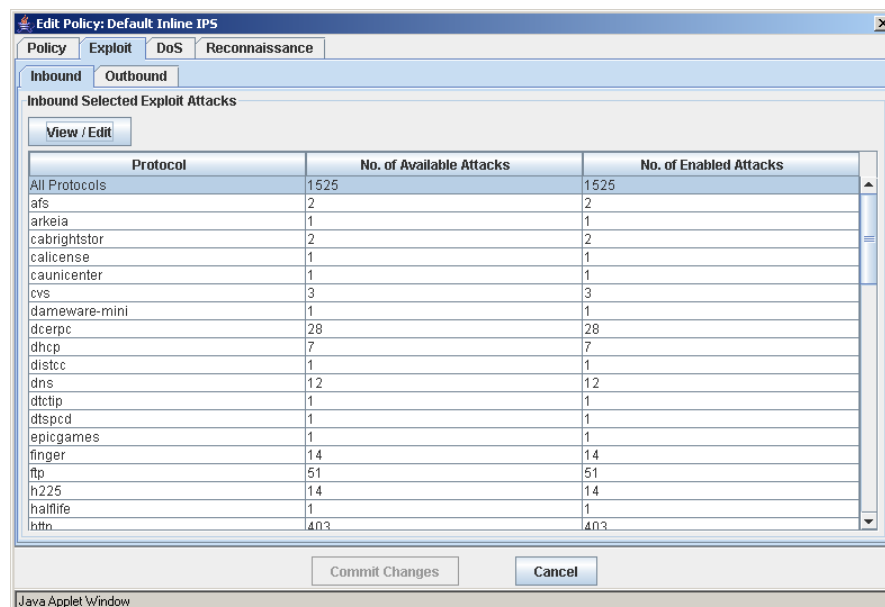


Figure 7 - IntruShield: Policy Editor

Once the rule set has been created it can be applied to a policy, and different rule sets can be applied to inbound traffic and outbound traffic if required. Whilst this might seem overkill, it is actually an incredibly powerful feature that allows the administrator to refine the differences in how inbound and outbound traffic are inspected. Whereas most will assign the same rule set in both directions, some will undoubtedly find this capability invaluable.

Once one or more rule sets have been applied, the Policy Editor displays the number of attacks that have been selected, grouped by protocol. Double clicking on a protocol brings up a list of the individual attacks, from where it is possible to customise certain elements of those attacks, including:

- *Whether the attack is enabled or disabled*
- *Attack severity*
- *Logging behaviour - the sensor can log the entire packet or a specific number of bytes, and in addition can log 256 bytes of application data prior to the attack*
- *Duration of logging - the sensor can capture the attack packet only, capture a specified number of packets, for a specified length of time, capture the entire flow, or deliver forensic logging where all subsequent communication between attacker and victim can be logged.*

- *Sensor actions - send TCP reset (to source, destination or both), send “ICMP Host Not Reachable” packet to source, reconfigure the firewall, drop attack packet and all subsequent packets for that flow (this last option only works when sensor is in in-line mode, of course, whereas the remaining options can be employed when the sensor is operating in SPAN or tap mode).*
- *Alert filter - enables the administrator to suppress certain alerts from or to specific IP addresses or a range of addresses*
- *Notifications - e-mail, pager or script notifications from the ISM to the administrator*

A useful bulk-edit capability allows the administrator to select multiple attacks (using standard Windows selection capabilities via the SHIFT and CTRL keys) and subsequently apply changes to certain parameters (such as the enable/disable flag, or alert response) to an entire group of attacks in a single operation.

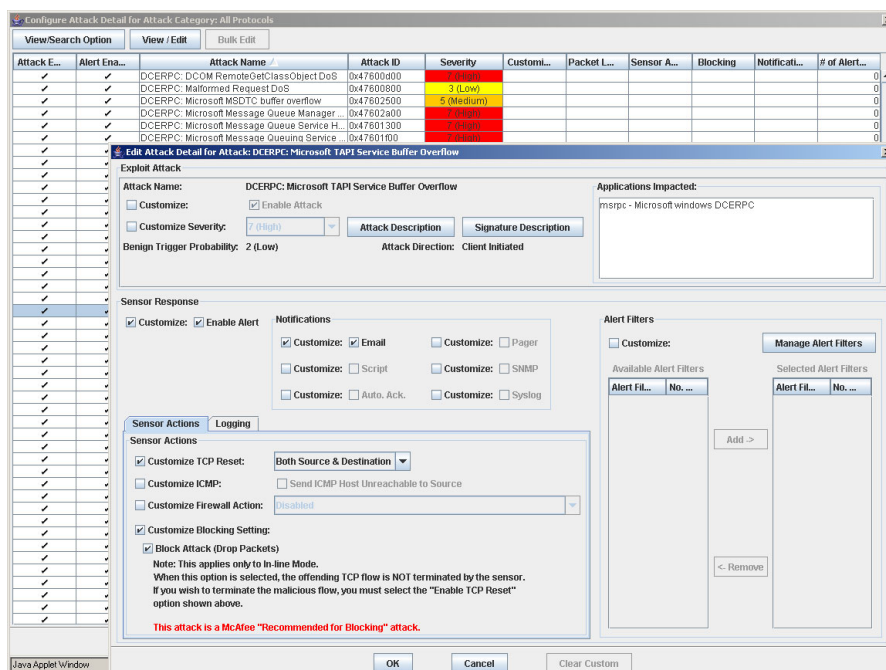


Figure 8 - IntruShield: Editing attack details

Note that it is impossible to tune the actual built-in signatures in any way. This means that if a built-in signature is misfiring for some reason, it is not possible to alter the regular expression, for example, to ensure that it does not trigger on benign network traffic. Instead, an administrator can employ alert filters or disable the misfiring attack signature, after which he would have to start from scratch in creating his own user-defined signature.

A powerful search facility allows the administrator to display all attacks, only those recommended by McAfee for blocking, or by a number of user-defined search options, and bulk edit operations can be applied to the results of the search.

Whereas the bulk edit capability allows the administrator to change multiple attacks in a single policy, there is also the requirement to change one or more attacks across **all** policies (to change an attack response to *block* on every sensor in an organisation which includes that attack in its policy, for example). This bulk edit across all policies capability is provided by the *Global Attack Response Editor* (GARE), and is something we would like to see in more products.

Separate policy settings are available for DoS attacks and reconnaissance probes. The sensor is automatically in *learning mode* by default, allowing it to monitor normal network traffic for a period of time so that it is able to determine what constitutes an abnormal flood. Individual DoS profiles can be created per sensor, and these can be uploaded to the ISM from where they can be applied to other sensors if required.

For those administrators who would prefer to have more manual control over the DoS detection process, it is also possible to switch to *threshold mode*, where he can set the threshold level and interval for individual DoS attacks. Note that both *threshold* and *learning mode* can be enabled or disabled on a per-attack basis, and both methods worked extremely well in our tests.

Finally, reconnaissance probes - port scans, host sweeps and brute force password cracking attempts - are configured purely on a per sensor basis and cannot be a part of a global policy. Each of these are configured using thresholds, and once again the default settings worked well in our tests.

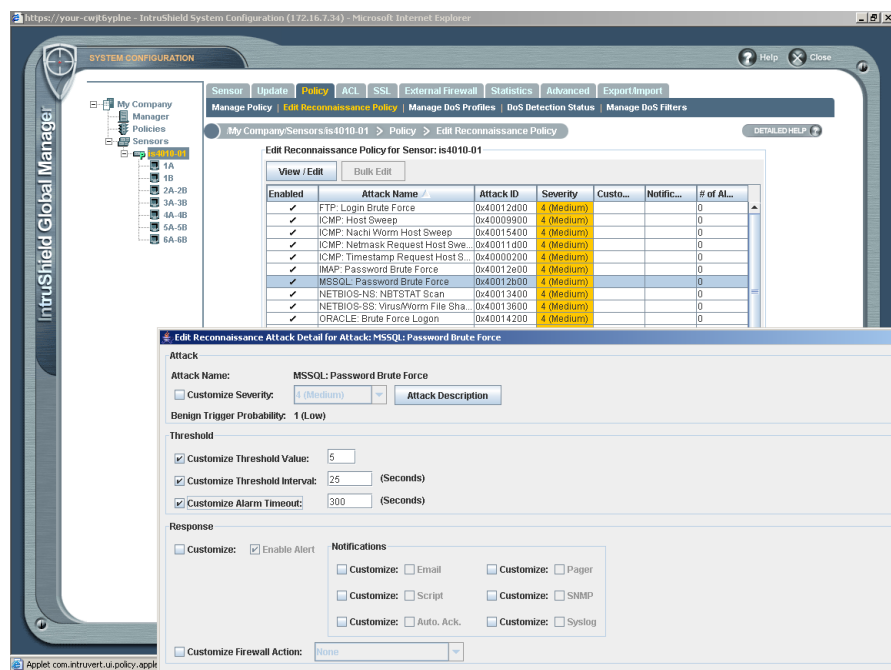


Figure 9 - IntruShield: Reconnaissance policy

After a policy has been configured using the Policy Editor, it can be applied to a resource - for example, an entire admin domain (or child domain), a sensor, an individual interface, or a sub-interface. Once applied to a resource, the policy is then propagated to the appropriate sensor (or sensors) for enforcement.

Whenever a policy is amended, the *Sensor Update* page provides an at-a-glance list of all the Sensors which require updating as a result, with a check box against each one. Most of the time, all that is required is to click on the *Update* button, but it is a simple matter to uncheck any Sensors to which the administrator does not wish to apply the changes made.

We found it to be fairly slow to deploy a policy to one or more sensors, although it should be noted that there is no interruption in traffic processing at any point in the update process. Whenever new signature updates are downloaded from the Web, the new signatures are added to *all* policies (within the confines of the applied rule sets), which can also be quite slow.

IntruShield's *Virtual IPS (VIPS)* feature enables the administrator to configure multiple policies for multiple unique environments all monitored with a single IntruShield sensor. Different policies can be applied to different interfaces, for example, allowing one pair of interfaces to monitor the DMZ with a predominantly Web-based policy in in-line mode, whilst another interface monitors the internal network in SPAN mode using the *Default IDS* policy.

A clever feature of IntruShield sensors takes port monitoring one step further than the physical interface-level, however. Using sub-interfaces, it is possible to segment the security management and apply policies at a traffic sub-flow level within an interface - in other words, the sensor monitors only a portion of the traffic passing through a physical interface. This sub-interface is also known as a *Virtual IPS (VIPS)*.

A VIPS can be defined based on a block of IP addresses (a CIDR block), or on one or more VLAN tags. IntruShield sensors can process these segments of data and apply multiple traffic policies for the multiple subnets transmitting across a single wire, right down to policies protecting individual hosts.

Other IPS products may allow filtering of addresses to achieve something similar, but they still generally only allow a single policy to be applied to each sensor. The particularly clever feature of IntruShield is the ability to support up to 1000 VIDS per sensor, and a different policy can be applied to every single VIDS. This means that on some networks, it would be possible to have a separate unique detection policy running for every single host on the network, all monitored from a single sensor!

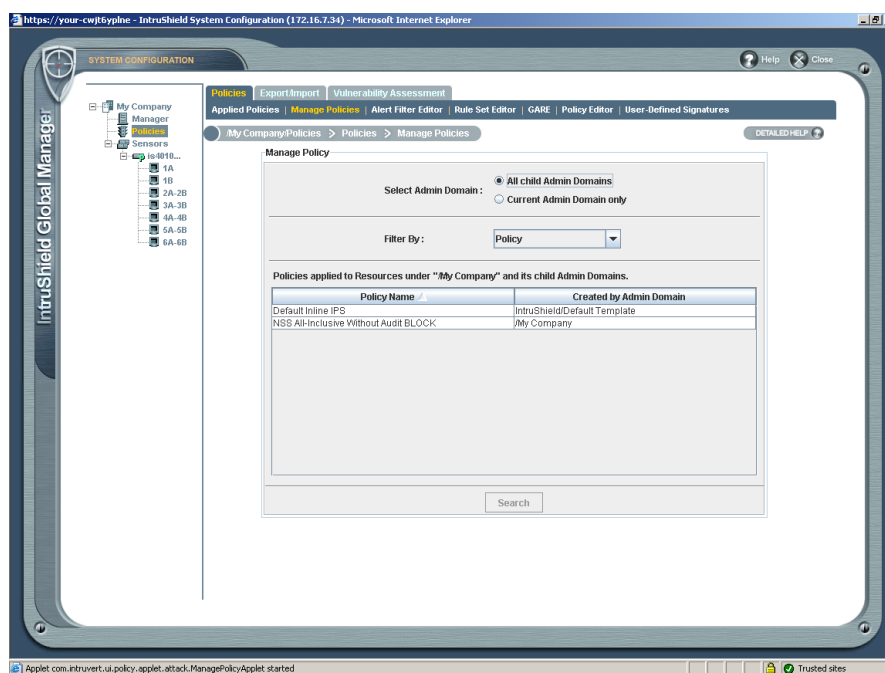


Figure 10 - IntruShield: Managing policies

It is also possible to mix interface modes in a single sensor, and apply different policies to each interface. One scenario we tested, for example, was to use two ports paired in in-line mode running a restrictive policy with intrusion prevention enabled for certain signatures, but with all the signatures with potential for false alarms disabled.

We also created a VIPS on the same interface pair which consisted of a single mail server, to which we applied a slightly different in-line policy with more focus on SMTP exploits.

We then used one of the remaining ports in SPAN mode attached to the SPAN port of our switch, and with a much broader policy applied which was set to capture the entire flow. Although the initial VIPS configuration requires some thought, once it has been accomplished, assigning policies to the separate VIPS is extremely simple, and the whole test scenario worked perfectly.

Ports and sub-interfaces are managed via the *Resource Tree* in the *System Configuration* utility, where it is possible to create sub-interfaces beneath an interface node, or VIPS nodes within child domains - both of these are different manifestations of the Virtual IPS, and allow the allocation of multiple policies to the same physical interface.

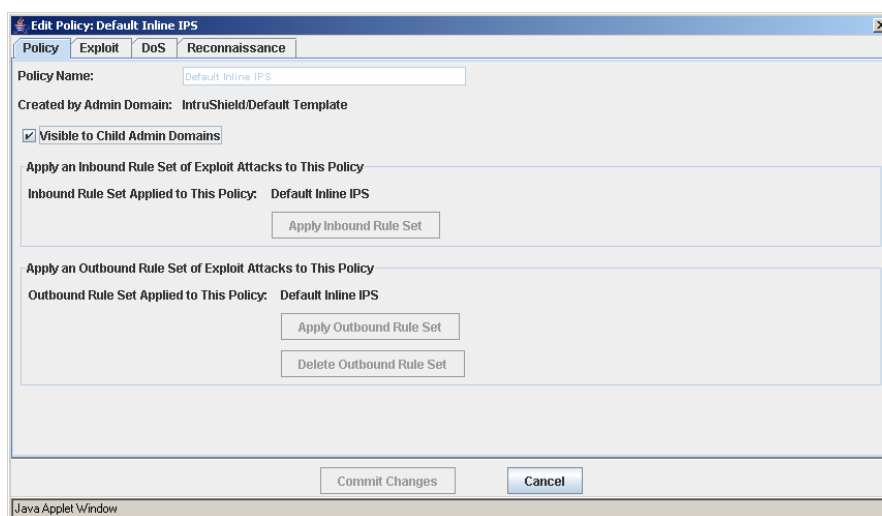


Figure 11 - IntruShield: Assigning rule sets and access to policies

A VIPS within a child domain can be allocated to an administrator who only has rights to that child domain and nothing else. Thus, when that administrator logs in, he will be able to configure and allocate policies to the VIPS under his control without affecting any other interfaces or VIPS in the system.

One area that does require improvement in the current release is that when creating a policy it is possible have it visible to **all** child domains or **none**. In a Managed Services environment, the root-level administrator may want to have individual policies visible only to the administrators in a particular domain. Right now every administrator can view every policy, and even though the granular admin roles would prevent them from applying policies to Sensors over which they have no control, the fact that the administrator of one client can see the policy used by another client would be cause for concern to some.

In our opinion, this is the **only** feature which prevents this being **the** perfect management system for Managed Service Providers, and McAfee has promised to address this with full-blown per-policy ACLs in a future release.

Alert Handling

Alerts exist in one of three states within the IntruShield system: *unacknowledged*, *acknowledged*, and *marked for deletion*.

When an alert is first raised, it appears in the ISM Console in an *unacknowledged* state, and remains in that state until the administrator either acknowledges or deletes it.

These alerts display in the *Unacknowledged Alert Summary* section of the Network Console and the *Real-time view* in the Alert Viewer. Acknowledging alerts dismisses them from these views, after which they display only in the *Historical* view in the Alert Viewer and in reports. It is not possible to annotate alerts as they are acknowledged, however - a shame, since that would be a useful facility to record the results of an investigation into the alert and why it was eventually regarded as unimportant.

Deleting an alert both marks it for deletion and acknowledges it in the same operation. The alert is not actually deleted until a scheduled *File Maintenance* operation takes place, however, at which time IntruShield removes all alerts marked for deletion along with any alerts meeting the deletion criteria specified in the scheduler (older than 30 days, for example).

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

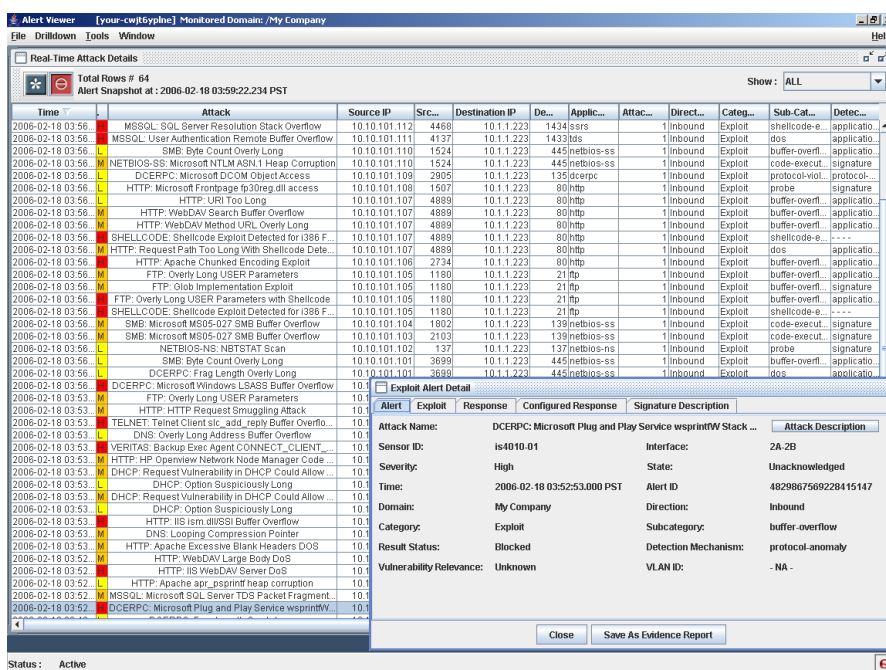


Figure 12 - IntruShield: The Alert Viewer

The IntruShield *Alert Viewer* is available via a hyperlink on the *Network Console* display. When it is first opened, the administrator must specify a time frame and a type of view - either *Real-time* or *Historical*. The *Real-time View* sets the alert filter to display information retrieved from an “alert cache” for a configured number of unacknowledged alerts. Once opened, the *Real-time View* refreshes frequently to display the alerts that are being detected by the sensors, but once closed, the only way to report on those alerts is to re-open the *Viewer* in *Historical* mode.

The alert cache stores up to 500,000 of the most recent unacknowledged alerts, and all cached alerts are also listed in the database.

Since the cache is a FIFO system, the oldest alerts in the cache are dropped once the cache maximum has been reached with newer incoming alerts that “overflow” the cache. Dropped alerts are simply discarded, since they have also been written to the database and are thus a matter of permanent record. At any point, the administrator can freeze the real-time display in order to work on the alerts currently on view (re-sorting columns, and so on).

The Historical View sets the filter to retrieve information for both acknowledged and unacknowledged alerts written to the database within a specified time frame. Older data can be archived to separate files based on date range, and can be restored to any ISM server for further analysis at a later date.

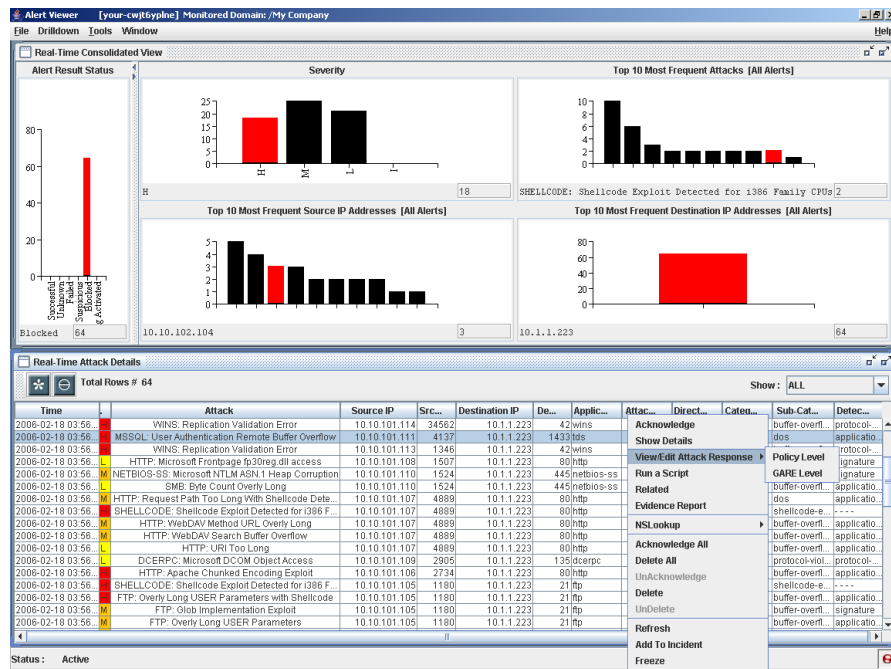


Figure 13 - IntruShield: Consolidated Alert View

Once alerts have been retrieved either from a particular time period, or in real time, the Alert Viewer interface displays the alerts in four main views:

- **Alerts By Time View** - the number of alerts in time intervals that have occurred in the last two hours (Real-Time) or a desired time frame (Historical). The Alerts By Time View displays a bar graph with the number of alerts that have occurred in the specified time frame. Each bar contains information related to the number of alerts and a time frame in which the alerts occurred.
- **Consolidated View** - displays alerts split into five panes (categories) for statistical review. Each pane is a bar graph, and each bar represents several alert instances grouped by a specific parameter. An alert may appear in a bar in more than one pane if that alert has met the statistical parameters of multiple categories. The categories are:
 - **Alert Result Status:** Lists alert totals by result: Successful, Unknown, Failed, Suspicious, or Blocked.
 - **Severity:** Lists alert totals by severity level: High, Medium, or Low.
 - **Top 10 Attacks:** lists the top 10 attacks by number of triggered alerts.

- **Top 10 Source IP Addresses:** lists the 10 most common source IP addresses by number of triggered alerts.
- **Top 10 Destination IP Addresses:** lists the 10 most-targeted destination IP addresses by number of triggered alerts.
- **Alert Details View** - displays all of the alerts, sorted by order of occurrence, for the selected time span. Alert details are presented in columns which display packet fields such as source and destination IP, as well as sensor analysis fields such as attack severity and type.
- **All Attacks View** - contains a list of all attacks seen in the set of alerts currently loaded into the Alert Viewer, as well as their associated alert and attack counts. Static information is also displayed for each attack type, such as severity, benign trigger probability, category, sub-category, detection mechanism, attack description and application protocol.

The real power of the Alert Viewer lies in the drill-down capabilities. Right-clicking or double-clicking on any column in a graph in the *Consolidated View* (or using the *Drilldown* toolbar menu) will cause the remaining graphs to be redrawn with new data.

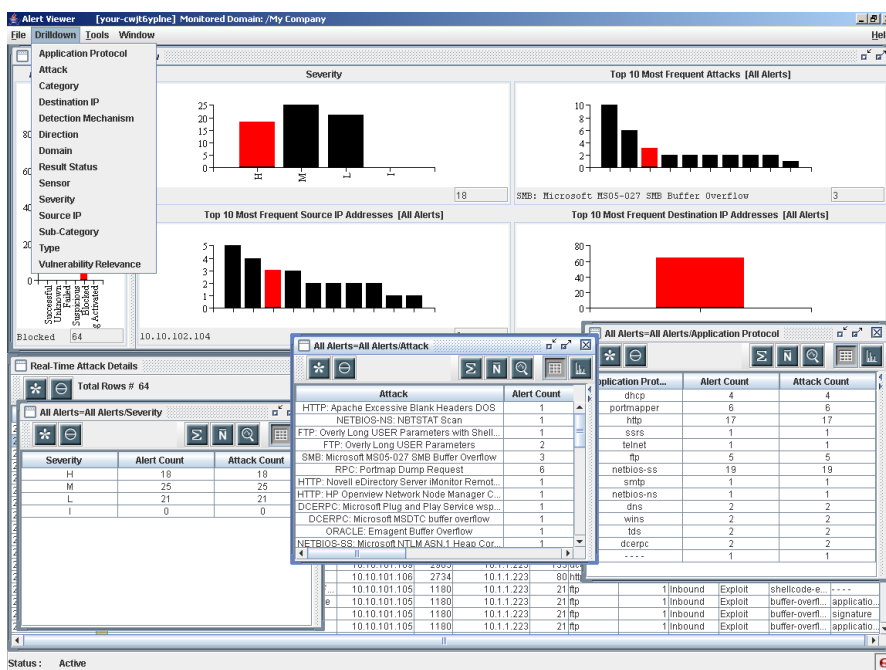


Figure 14 - IntruShield: Consolidated Views with drill down

For example, initially, the *Top 10 Alerts* graph shows the top 10 alerts overall. Double click on the *High* column in the *Severity* graph, however, and the *Top 10 Alerts* chart is redrawn to display the top 10 alerts which have a *Severity* level of *High*. Subsequently double-clicking on one of the top 10 bars launches an additional window containing a list of all the individual alerts - the *Alert Details View*. It is also possible to right-click on a bar in one of the *Consolidated View* graphs and bring up a summary window grouped and sorted by time, severity, attack, IP address, interface, protocol, domain, type, or sensor.

Once again, any of the groups can be double-clicked to call up the *Alert Details View* populated with the individual alert entries for that group (or right-clicked, and the entire group of alerts acknowledged or deleted in a single operation).

There are thus many different ways of slicing and dicing data in order to drill down from very high level summaries to very low level detail. Should the number of open “floating” windows become confusing, a useful *Window Manager* provides the means to see an overview of open windows and jump quickly between them. It is also possible to save window layouts in user preferences to preserve them each time the Console is started, and save selected windows in PDF or CSV format to create quick reports. Overall the system works extremely well.

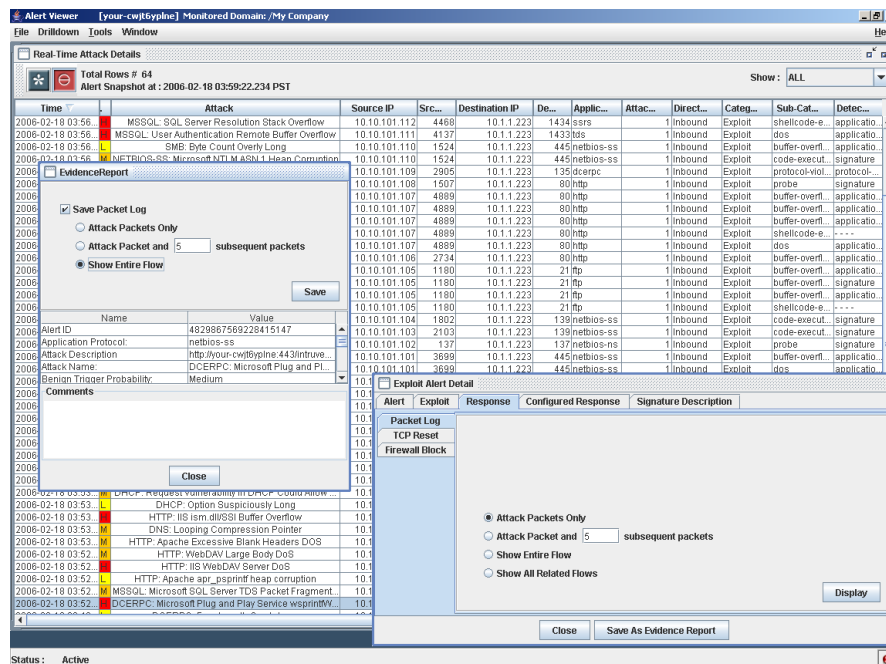


Figure 15 - IntruShield: Viewing Exploit Alert Details

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

Note that by processing network traffic passing through a sensor, it is sometimes possible for the IntruShield to determine the results status of an attempted attack - this can be one of *failed*, *successful*, *unknown*, *suspicious* or *blocked*. Of course, this works with varying degrees of success depending on the exploit and subsequent traffic.

Many of our tests cases showed up simply as “*Unknown*”, but a significant number did show as “*Suspicious*”, and those which had very obviously failed (i.e. if a 404 error was returned by the Web server) or succeeded (i.e. if a Unix command shell was detected following the exploit) were always correctly identified. This is a useful capability.

The *Alert* tab provides details specific to the attack type, which could be an *exploit*, a *DoS attack*, a *port scan*, and so on. This tab includes information such as source and destination IP address, network protocol, application protocol, threshold values, and target ports. The *Response* tab provides response information, where the administrator can examine the exploit packet, including the 256 bytes immediately preceding it or even the entire flow if those logging options were enabled for that particular signature.

A packet log is created by an IntruShield sensor capturing the network traffic of and around an offending transmission. If logging was enabled for specific exploit attacks (the *Configured Response* tab provides details on exactly which responses were triggered by the alert, and thus provides an indication to the administrator of exactly what logging data is available), the appropriate packet logs are saved in library packet capture (libpcap) format, and stored in the ISM database.

Ethereal is used as the packet viewer, which is actually a very sensible option since so many security personnel will be familiar with it, and it also provides excellent protocol decode capabilities.

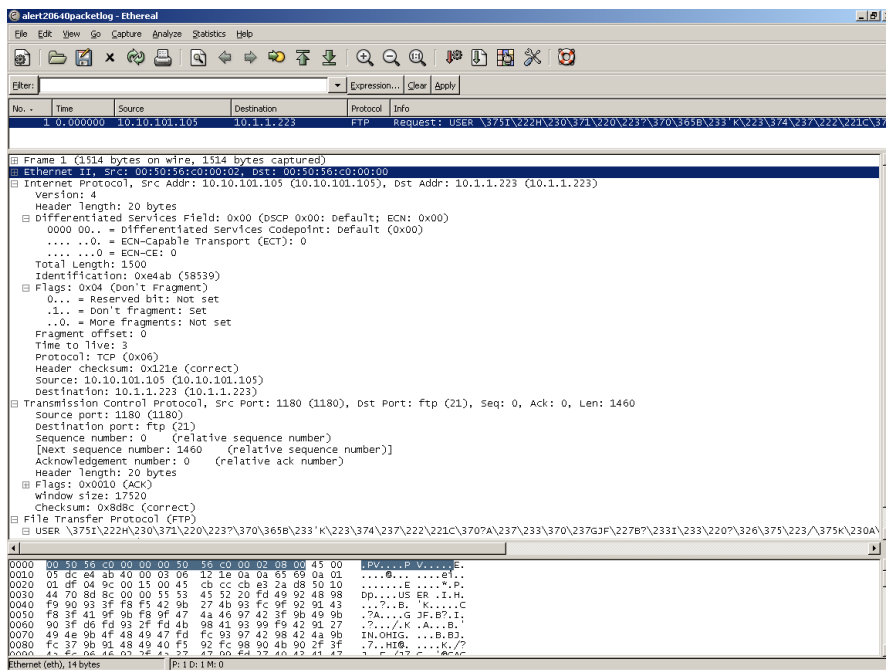


Figure 16 - IntruShield: Using Ethereal to view packet details

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

Once an alert has been examined and investigated, it can be acknowledged. At that point it is removed from the various statistical/summary views, and is subsequently only retrieved from the database for *Historical View* searches and *IPS Reports*. An alert can also be saved as an *Evidence Report*. This opens a complete view of a selected alert row in a separate window, and provides the option to save the alert information, including a packet log (if available), in a zip file which can be saved or passed to others for forensic analysis.

For more extensive forensic analysis, IntruShield provides the concept of “*Incidents*”. An Incident is a collection of related alerts which can be correlated automatically by the *Incident Generator* based on pre-configured criteria such as 100 attacks from the same source in 15 minutes. This is accomplished by a process known as the Incident Generation Service which runs in the background on the ISM server.

Incidents can also be created manually by the administrator, who can select and group together a collection of related alerts from the Alert Viewer using the *User-Generated Incidents* tool.

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

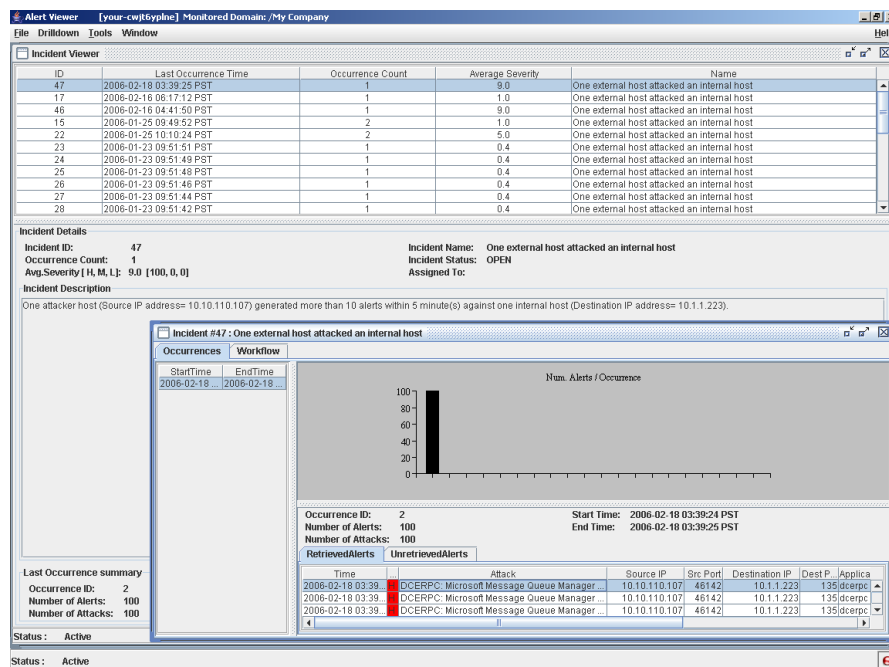


Figure 17 - IntruShield: Incident Viewer

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

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

Reporting and Analysis

Whilst the *Alert Viewer* provides both real-time monitoring and interactive forensic analysis capabilities (as well as the ability to save selected windows in PDF/CSV format to create quick reports), the *Report Generator* offers the administrator the opportunity to create more comprehensive, and higher-level summary reports both in text and graphical format. As with the *Alert Viewer*, the *Report Generator* is available at the click of a button from the main IntruShield *Network Console* view.

The *IPS reports* (still labelled as *IDS Reports* on the menu) provide summary information on the alerts generated from the installed sensors. The generated alert information can include source and destination IP of the attack, time when attack occurred, sensor that detected the attack, and so forth.

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

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

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

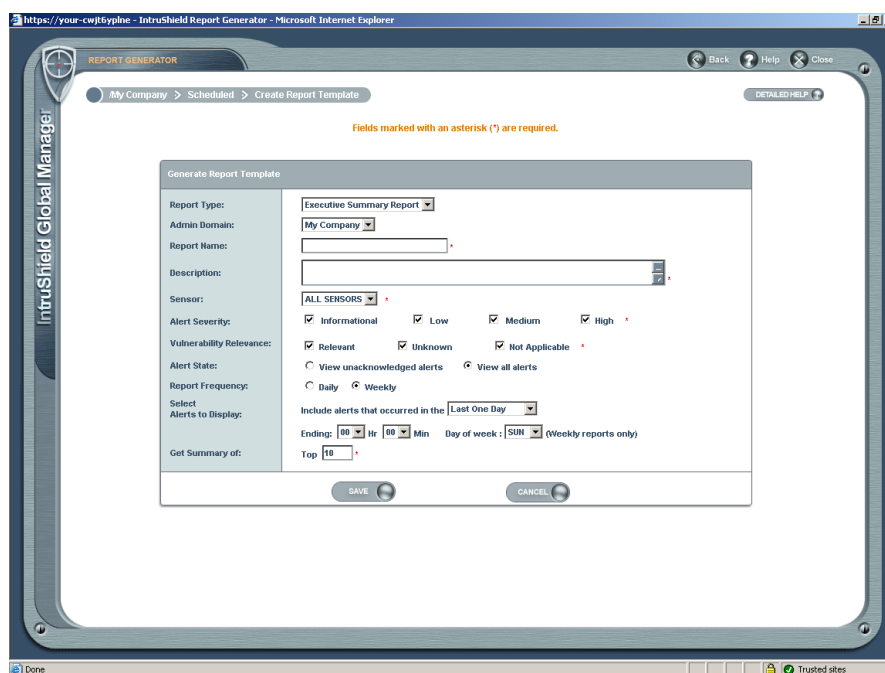


Figure 18 - IntruShield: Creating report templates

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

- **Version Report** - Information on the versions of software and signatures in use.
- **Admin Domain Configuration Report** - Information on the admin domains and users controlled through the ISM server.

- **Manager Configuration Report** - Information on the configured Notification Mail Server and the Proxy Server.
- **Sensor Configuration Report** - Information on the current software/signature versions running and the status of a sensor's ports.
- **ACL Configuration Report** - Provides a detailed view of the ACL rules configured for one or more sensors
- **Sensor Policy Configuration Report** - Information on the policies applied to one or more sensors.
- **Admin Domain Policy Configuration Report** - Information on the policies applied to one or more admin domains.
- **IDS Policy Report** - Information on all of the IDS/IPS policies available.
- **Rule Set Report** - Information on all of the rule sets available.
- **Alert Filter Report** - Information on all of the alert filters available for policy application.
- **Audit Report** - Information on the actions performed by IntruShield users.
- **Fault Log Report** - Information on ISM and Sensor fault logs

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

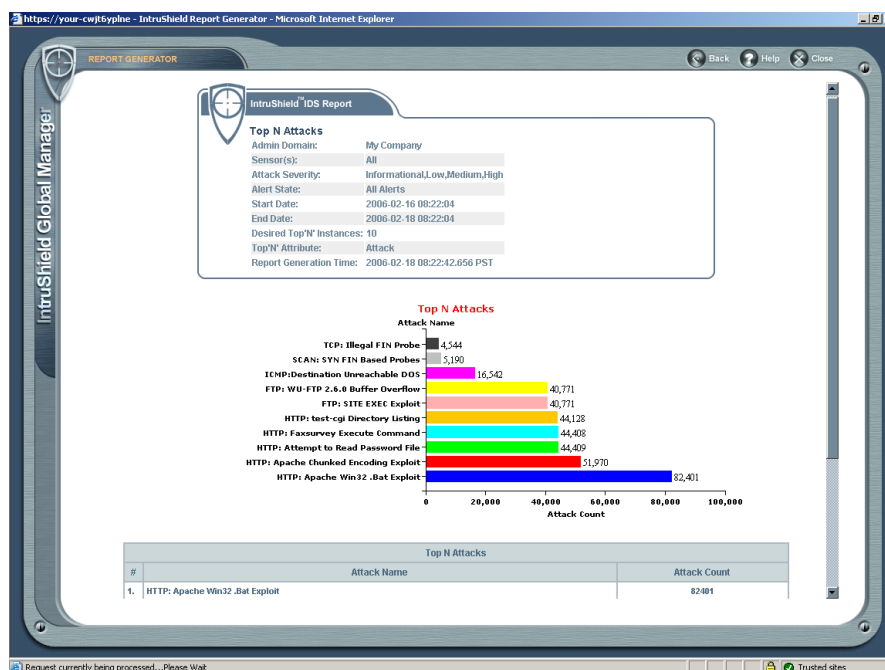


Figure 19 - IntruShield: Top n Report

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

Verdict

Performance

Note that although MacAfee rates the IntruShield 4010 as a 2Gbps device, it was tested to a maximum of 1Gbps in this round of testing.

As you would expect with this amount of headroom, performance at all levels of our 1Gbps load tests was impeccable, with 100 per cent of all attacks being detected and blocked under all load conditions.

We also ran some tests up to the maximum of 2Gbps, and under normal network conditions, we would have no hesitation in rating the IntruShield 4010 as a true 2Gbps device.

Latency figures were excellent across the board with all packet sizes (even down to 64 byte packets) and all traffic loads, and behaviour throughout the tests with no background traffic was very predictable, with minimal increases in latency as traffic levels increased from 250Mbps to 1Gbps across each packet size.

Placing the device under a half load of 500Mbps of HTTP traffic, we noted significant increases in latency, although all results remained below the “magic figure” of 300 microseconds - our limit for deploying in-line devices in the core of the network. HTTP response times were also very good, meaning IntruShield could be situated anywhere on a Gigabit network, either internally or at the perimeter.

Using the SYN proxy capability, SYN Flood mitigation was almost total, and 100Mbps of SYN flood traffic had a minimal effect on latency of normal traffic through the device. Average HTTP response increased from 204ms to 1716ms during the SYN flood, although no legitimate traffic was blocked during the attack.

The IntruShield 4010 performed consistently and completely reliably throughout our tests, and exposing the sensor interface to extreme levels of ISIC-generated traffic had no adverse effect.

Security Effectiveness

Signature recognition and blocking performance were excellent out of the box (95 per cent), and was increased to a perfect 100 per cent after the application of a signature pack update which was provided to us in under 24 hours.

We noted a minimum of “noise”, with very few test cases raising multiple alerts for a single exploit, and the accuracy of the exploit descriptions was high. Performance in our “false negative” tests was very good out of the box, and there is every indication that the majority of signatures are written for the underlying vulnerability rather than specific exploits.

IntruShield’s resistance to false positives was excellent in our tests, and was increased to a perfect 100 per cent following the signature update.

IntruShield arrives with a number of default policies configured for different environments and with sensible PASS and BLOCK actions set for appropriate signatures.

Resistance to known evasion techniques was excellent, with IntruShield achieving a clean sweep across the board in all our evasion tests.

The IntruShield 4010 demonstrated perfect performance in our stateful operation tests out of the box, handling over 1 million open connections with no tuning required.

Usability

The current UI is beginning to look a little dated, but a sneak glimpse at the new version currently in development (due to be released later in 2006) impressed us - the new look is much cleaner and more modern, whilst retaining much of the current mode of operation familiar to existing users.

The IMS and Console have been designed from the ground up to handle large distributed deployments and even managed services environments, and they contain several useful features to make this type of deployment easier to handle.

To begin with, the ability to define up to 1000 Virtual IPS' across the twelve ports and assign an individual policy to each of them makes this one of the most flexible systems we have seen in our labs - IntruShield was the first with this, and only recently has the competition begun attempts to emulate it. That flexibility is boosted by the fact that each port or port pair can be configured in different ways - either in "traditional" SPAN mode, or in tap mode, or in in-line mode for the ultimate in protection. Or they can be grouped together as a port cluster and the traffic aggregated across them.

The admin domains and user roles make it easy to delegate the most fine-grained control across the largest organisation. The rule-based policy definition makes it easy to define complex policies, which can then be rolled out to a single sensor or the entire network by simply allocating the policies at the appropriate level in the Resource Tree. And once the policies have been applied, the alert handling (including correlation) and forensic analysis capabilities are incredibly powerful and flexible.

The internal firewall is also a useful feature, providing the means for an administrator to apply firewall rules in the core of the network without having to deploy a layer 3 device.

Finally, the SSL decryption capability is unique at the time of writing and enables IntruShield to decrypt selected SSL traffic on the fly and apply detection and protection capabilities to that traffic.

Overall, IntruShield is an excellent product which demonstrates high levels of performance and scalability, and incredibly flexible and powerful management and alert-handling capabilities.

Contact Details

Company: McAfee, Inc.

E-mail: john_vecchi@mcafee.com

Internet: www.mcafee.com

Address:
3965 Freedom Circle
Santa Clara, CA
USA

Tel: +1 800 338 8754

APPENDIX A – TEST RESULTS

The aim of this procedure 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 Cisco 6500-Series switches (these have a mix of fibre and copper Gigabit interfaces). All devices are expected to be provided as appliances - if software-only, the supplier pre-installs the software on the recommended hardware platform. The sensor 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 Spirent Smartbits *receive* port - is connected to the internal network. The device under test is connected between two “gateway” switches - one at the edge of the external network, and one at the edge of the internal 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 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 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. These are updated/changed for every new test, and all exploits are run with no load on the network and no IP fragmentation.

Our attack suite contains over 100 basic exploits (plus variants) covering the following areas:

- [Test 1.1.1 - Backdoors \(standard ports and random ports\)](#)
- [Test 1.1.2 - DNS/WINS](#)
- [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](#)
- [Test 1.1.15 - Voice](#)

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 than the underlying vulnerability, or if it relies purely on pattern matching, some of these false alarms could be alerted upon.

The product 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 sensor could block legitimate traffic inadvertently.

- [Test 1.2.1 - False positives](#)

Section 2 – 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. Note that common/older attacks have been chosen deliberately for this particular test to ensure that ALL products tested have signatures in place for the evasion tests.

- [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*
- *Test 2.2.20 - TCP segmentation - ordered 16 byte segments, segment overlap (favour new (Unix))*

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/passwords for backdoors](#)
- [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 \(MS-RPC and Sun\)](#)
- [Test 2.4.7 - HTTP exploits to non-standard port](#)

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 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 - 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 many flooding tools 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 (although “mid-flow” alerts are permitted).

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. This test is run using the default sensor settings (no tuning of sensor parameters).

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 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 sensor). The initial HTTP session is then completed with the second half of the exploit and the session is closed. If the sensor is still maintaining state on the 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 - a product will fail the test if it fails to generate an alert after the second packet is transmitted, or if it raises an alert on either half of the exploit on its own.

At each step, we ensure that the sensor 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). We then launch further exploits whilst the Avalanche/Reflector devices “churn” connections at the maximum level set, ensuring that the sensor is still capable of detecting and blocking freshly-launched exploits as old connections are torn down and new ones recreated constantly.

- [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 device under test. 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 sensor 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 sensor 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 sensor (bi-directionally, maximum of 1Gbps).

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. The aim of this test is purely to determine the raw packet processing capability of the sensor, and its effectiveness at passing “useless” packets quickly in order to pass potential attack packets to the detection engine. The range of packet sizes has been selected to mirror the maximum, minimum and average packet sizes used in our HTTP stress tests.

- **Test 4.1.1 - 256 byte packets - maximum 453,000 packets per second:** *This test is roughly equivalent to a 40,000 connections per second test in our HTTP stress tests (in terms of packet size and packets per second rate), and has been included to provide an indication of the packet processing performance under the most extreme conditions for most devices - it is unlikely that any real-life network will ever see network loads of over 450,000 256-byte packets per second unless under severe DOS conditions. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic.*
- **Test 4.1.2 - 550 byte packets - maximum 220,000 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 - 1000 byte packets - maximum 122,000 packets per second:** *This test is the complete opposite of the 256 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 1000 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 (or larger) 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 sensor.

The use of multiple Spirent Communications **Avalanche 2500** and **Reflector 2500** devices allows us to create true “real world” traffic at speeds of up to 4.2 Gbps as a background load for our tests. Our Avalanche configuration is capable of simulating over 5 million users, with over 5 million concurrent sessions, and over 200,000 HTTP requests per second.

By creating genuine session-based traffic with varying session lengths, the sensor 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 1000 bytes - maximum 120,000 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 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 225,000 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 sensors to perform well in this test.
- **Test 4.2.3** - Max 10,000 new connections per second - average packet size 440 bytes - maximum 275,000 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 sensor, and represents a very heavily used production network.
- **Test 4.2.4** - Max 20,000 new connections per second - average packet size 360 bytes - maximum 320,000 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 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 225,000 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 sensors to perform well in this test.
- **Test 4.3.2** - Max 10,000 new connections per second - average packet size 440 bytes - maximum 275,000 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 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 (540 byte packets) + 20% FTP traffic + 6% UDP traffic (256 byte packets). Max 4000 new connections per second - average packet size 540 bytes - maximum 215,000 packets per second - maximum 750 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 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-Xeon server with Gigabit interface and 4GB RAM. This server holds a copy of The NSS Group Web site, and is capable of handling a full 1Gbps 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 **20 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 slightly more “bursty” in nature.

- **Test 4.5.1 - Pure HTTP Traffic (simulated browsing session on NSS Web site):** Max 4700 new connections per second - 20 new users per second - average packet size 560 bytes - maximum 210,000 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 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 3700 new connections per second - average packet size 560 bytes - maximum 205,000 packets per second - maximum 1,500 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 sensors to perform well at all load levels in this test.*

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 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. We also determine the effect of high levels of normal HTTP traffic and a basic DOS attack on the average latency and user response times.

Test 5.1 - Latency

We use Spirent SmartFlow software and The Smartbits SMB6000 with Gigabit TeraMetrics cards to create multiple traffic flows through the 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 (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 (256 bytes, 550 bytes and 1000 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 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 112,500 packets per second).
A 250Mbps bi-directional load of SmartFlow traffic at various packet sizes (256 bytes, 540 bytes and 1000 bytes) 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 bi-directional load of SmartFlow traffic at various packet sizes (256 bytes, 540 bytes and 1000 bytes) is then passed across the infrastructure switches and through the device and the packet loss and average latency are recorded. The device should be configured to detect/block/mitigate the DOS attack by the most efficient method available.

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 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 112,500 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 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 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 sensor 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 sensor 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 sensor, and the ISIC target directly to the internal interface. ISIC traffic is transmitted through the sensor (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.

Test 7.1.3 - *We note whether the ISIC attacks themselves are detected by the sensor even though targeted at the management port.*

McAfee IntruShield 4010 V3.1 Test Results

Section 1 - Detection Engine

Test 1.1 – Attack Recognition	Attacks	Default ARR	Default ARRB	Custom ARR	Custom ARRB
Test 1.1.1 - Backdoors	7	7	7	7	7
Test 1.1.2 - WINS/DNS	3	3	3	3	3
Test 1.1.3 - DOS	10	10	10	10	10
Test 1.1.4 - False negatives (modified exploits)	14	12	12	14	14
Test 1.1.5 - Finger	4	4	4	4	4
Test 1.1.6 - FTP	5	5	5	5	5
Test 1.1.7 - HTTP	43	41	41	43	43
Test 1.1.8 - ICMP	2	2	2	2	2
Test 1.1.9 - Reconnaissance	8	8	8	8	8
Test 1.1.10 - RPC	9	8	8	9	9
Test 1.1.11 - SSH	1	1	1	1	1
Test 1.1.12 - Telnet	1	1	1	1	1
Test 1.1.13 - Database	1	1	1	1	1
Test 1.1.14 - Mail	1	1	1	1	1
Test 1.1.15 - Voice	1	1	1	1	1
Total	110	105 / 110	105 / 110	110 / 110	110 / 110
		95%	95%	100%	100%

Test 1.2 – Resistance to False Positives	Default	Custom
Test 1.2.1 - Suspicious FTP traffic	PASS	PASS
Test 1.2.2 - HTTP "exploit" using incorrect method	FAIL	PASS
Test 1.2.3 - Retrieval of Web page containing "suspicious" URLs	PASS	PASS
Test 1.2.4 - Simple SMTP QUIT command	PASS	PASS
Test 1.2.5 - Normal NetBIOS copy of "suspicious" files	PASS	PASS
Test 1.2.6 - Normal NetBIOS traffic	PASS	PASS
Test 1.2.7 - POP3 e-mail containing "suspicious" URLs	PASS	PASS
Test 1.2.8 - POP3 e-mail with "suspicious" DLL attachment	PASS	PASS
Test 1.2.9 - POP3 e-mail with "suspicious" Web page attachment	PASS	PASS
Test 1.2.10 - SMTP e-mail transfer containing "suspicious" URLs	PASS	PASS
Test 1.2.11 - SMTP e-mail transfer with "suspicious" DLL attachment	PASS	PASS
Test 1.2.12 - SMTP e-mail transfer with "suspicious" Web page attachment	PASS	PASS
Test 1.2.13 - SNMP V3 packet with invalid parameter	PASS	PASS
Test 1.2.14 - Fake DNS /bin/sh buffer overflow	PASS	PASS
Test 1.2.15 - Inter-firewall communication traffic	PASS	PASS
Test 1.2.16 - Fake SQL Slammer traffic	PASS	PASS
Test 1.2.17 - File copy of GIF file (contains bytes which look like NOP sled)	PASS	PASS
Total Passed	16 / 17	17 / 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	YES	YES
Test 2.1.3 - FTP CWD root	YES	YES
Test 2.1.4 - ISAPI printer overflow	YES	YES
Test 2.1.5 - Showmount export lists	YES	YES
Test 2.1.6 - Test CGI probe (/cgi-bin/test-cgi)	YES	YES
Test 2.1.7 - PHF remote command execution	YES	YES
Total	7 / 7	7 / 7

Test 2.2 – Packet Fragmentation/Stream Segmentation	Detected?	Decoded?	Blocked?
Test 2.2.1 - IP fragmentation - ordered 8 byte fragments	YES	YES	YES
Test 2.2.2 - IP fragmentation - ordered 24 byte fragments	YES	YES	YES
Test 2.2.3 - IP fragmentation - out of order 8 byte fragments	YES	YES	YES
Test 2.2.4 - IP fragmentation - ordered 8 byte fragments, duplicate last packet	YES	YES	YES
Test 2.2.5 - IP fragmentation - out of order 8 byte fragments, duplicate last packet	YES	YES	YES
Test 2.2.6 - IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse	YES	YES	YES
Test 2.2.7 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new)	YES	YES	YES
Test 2.2.8 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old)	YES	YES	YES
Test 2.2.9 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums	YES	YES	YES
Test 2.2.10 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags	YES	YES	YES
Test 2.2.11 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence nos. mid-stream	YES	YES	YES
Test 2.2.12 - TCP segmentation - ordered 1 byte segments, duplicate last packet	YES	YES	YES
Test 2.2.13 - TCP segmentation - ordered 2 byte segments, segment overlap (favour new)	YES	YES	YES
Test 2.2.14 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers	YES	YES	YES
Test 2.2.15 - TCP segmentation - out of order 1 byte segments	YES	YES	YES
Test 2.2.16 - TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits	YES	YES	YES
Test 2.2.17 - TCP segmentation - ordered 1 byte segments, segment overlap (favour new)	YES	YES	YES
Test 2.2.18 - TCP segmentation - out of order 1 byte segments, PAWS elimination (interleaved dup segments with older TCP timestamp options)	YES	YES	YES
Test 2.2.19 - IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery	YES	YES	YES
Test 2.2.20 - TCP segmentation - ordered 16 byte segments, segment overlap (favour new (Unix))	YES	YES	YES
Total	20 / 20	20 / 20	20 / 20

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	YES	YES
Total	9 / 9	9 / 9	9 / 9

Test 2.4 – Miscellaneous Obfuscation Techniques	Detected?	Decoded?	Blocked?
Test 2.4.1 - Altering default ports	YES	YES	YES
Test 2.4.2 - Inserting spaces in FTP command lines	YES	YES	YES
Test 2.4.3 - Inserting non-text Telnet opcodes in FTP data stream	YES	YES	YES
Test 2.4.4 - Polymorphic mutation (ADMmutate)	YES	YES	YES
Test 2.4.5 - Altering protocol and RPC PROC numbers	YES	YES	YES
Test 2.4.6 - RPC record fragging (MS-RPC and Sun)	YES	YES	YES
Test 2.4.7 - HTTP exploits to port <> 80	YES	YES	YES
Total	7 / 7	7 / 7	7 / 7

Section 3 - Stateful Operation

Test 3.1 – Stateless Attack Replay	Alert?	Blocked?	Pass/Fail
Test 3.1.1 - Stateless Web exploits	NO	YES ¹	PASS
Test 3.1.2 - Stateless FTP exploits	NO	YES ¹	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	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Test 3.2.2 - Attack Blocking	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Test 3.2.3 - State Preservation	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Test 3.2.4 - Legitimate traffic blocking	PASS	PASS	PASS	PASS	PASS	PASS	PASS

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	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²
Test 3.3.2 - Attack Blocking	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²
Test 3.3.3 - State Preservation	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²
Test 3.3.4 - Legitimate traffic blocking	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²	PASS ²

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 - 256 byte packet test - max 453,000pps	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	
Test 4.1.2 - 550 byte packet test - max 220,000pps	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	
Test 4.1.3 - 1514 byte packet test - max 122,000pps	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 1000 bytes - max 120,000 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 225,000 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 275,000 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 360 bytes - max 320,000 packets per second	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	

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 225,000 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 275,000 packets per second - 10 sec delay - max 100,000 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 + 6% UDP (256 byte packets). Max 4000 connections per second - ave packet size 540 bytes - max 215,000 packets per second - max 750 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 4700 connections per second - 20 new users per second - ave packet size 560 bytes - max 210,000 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 + 6% UDP (256 byte packets). Max 3700 connections per second - ave packet size 560 bytes - max 205,000 packets per second - max 1,500 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	256	62.69	63.51	63.60	63.71
	550	86.45	86.69	86.90	87.42
	1000	125.19	125.58	125.72	126.96
Test 5.1.2 Average latency (µs) with background traffic (500Mbps HTTP traffic, max 2500 connections per second - ave packet size 540 bytes - max 112,500 packets per second)	256	261.92			
	550	266.83			
	1000	288.71			
Test 5.1.3 Average latency (µs) when under attack (100Mbps SYN flood)	256	81.49			
	550	103.29			
	1000	170.00			

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 112,500 packets per sec)	1651023	0	203	1709	204
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 112,500 packets per sec PLUS 100Mbps SYN flood)	1651261	0	1547	1921	1716

Section 6 - Stability & Reliability

Test ID	Result
Test 6.1.1 - Blocking Under Extended Attack	100%
Test 6.1.2 - Passing legitimate traffic under extended attack	100%
Test 6.1.3 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC	PASS

Section 7 - Management Interface

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

Notes:

1. This is configurable
2. No tuning was necessary for this test. Maximum open connections is fixed at 1.1 million out of the box.

Section 1: Detection Engine

We installed one sensor with the latest signature pack, and utilised the “All inclusive With Audit” policy provided out of the box - this policy has every attack signature and every audit signature (almost 1800 in total) enabled.

Signature recognition (with blocking disabled) was excellent out of the box (95 per cent), and was increased to a perfect 100 per cent after the application of a signature pack update which was provided to us in under 24 hours. Blocking performance was identical throughout the tests.

We noted a minimum of “noise”, with very few test cases raising multiple alerts for a single exploit, and the accuracy of the exploit descriptions was high. Performance in our “false negative” tests was very good out of the box, and there is every indication that the majority of signatures are written for the underlying vulnerability rather than specific exploits. Specific exploit signatures are also included where appropriate, however, to provide more accurate identification for the administrator.

A major concern in deploying an IPS is the blocking of legitimate traffic. IntruShield’s resistance to false positives was excellent in our tests, raising just one alert in our false positive test cases, and was increased to a perfect 100 per cent following the signature update.

IntruShield arrives with a number of default policies configured for different environments and with sensible PASS and BLOCK actions set for appropriate signatures.

Section 2: IPS Evasion

Resistance to known evasion techniques was excellent, with IntruShield achieving a clean sweep across the board in almost all our evasion tests.

Resistance to known evasion techniques was excellent, with IntruShield achieving a clean sweep across the board in all our evasion tests. *IP packet fragmentation, TCP stream segmentation, URL obfuscation, shell-code mutation, FTP evasion and RPC record fragmentation (ONC and MS-RPC)* all failed to trick IntruShield into ignoring valid attacks.

Not only were the fragmented and obfuscated attacks blocked successfully, but all of them were decoded accurately as well.

Section 3: Stateful Operation

The IntruShield 4010 demonstrated perfect performance in our stateful operation tests out of the box. The device maintained state on 1 million open connections, successfully detecting our half-open exploit as it was completed. It also continued to detect and block new exploits as we maintained 1 million open connections, and no legitimate traffic was blocked throughout these tests.

No tuning was necessary in order to support this level of open connections, enabling the 4010 to display a perfect set of results out of the box.

Stateless “exploits” are not alerted upon (this is correct behaviour in order to be resistant to *Stick* and *Snot* tools) and mid-flows are blocked by default. It is possible to configure the device to permit mid-flows if required.

Default operation of the device is to age out “least used” connections when the state tables are full or resources are low - this behaviour is not configurable. This means that it is technically possible to evade the 4010 once the state tables are full, since it will allow attack traffic from aged-out connections at that point. It should be noted that such a method of evasion would be very difficult given that IntruShield does not simply age out the oldest sessions.

The alternative is to deny new connections, which has the result of blocking legitimate traffic when resources are exhausted. Where only one method is available, NSS favours the IntruShield method, but the ability to configure this behaviour would be preferable.

Section 4: Detection/Blocking Performance Under Load

Note that although McAfee rates the IntruShield 4010 as a 2Gbps device, it was tested to a maximum of 1Gbps in this round of testing. As you would expect with this amount of headroom, performance at all levels of our 1Gbps load tests was impeccable, with 100 per cent of all attacks being detected and blocked under all load conditions.

We also ran some tests up to the maximum of 2Gbps, and under normal network conditions, we would have no hesitation in rating the IntruShield 4010 as a true 2Gbps device.

Section 5: Latency & User Response Times

Latency figures were excellent across the board with all packet sizes (even down to 64 byte packets) and all traffic loads. Latency ranged from 63µs with 250Mbps of 256 byte packets, to 127µs with 1Gbps of 1000 byte packets.

Behaviour throughout the tests with no background traffic was very predictable, with minimal increases in latency as traffic levels increased from 250Mbps to 1Gbps across each packet size.

Placing the device under a half load of 500Mbps of HTTP traffic, we noted significant increases from 63µs to 263µs with 256 byte packets, from 86µs to 267µs with 550 byte packets, and from 125µs to 289µs with 1000 byte packets.

However, all results remained below the “magic figure” of 300 microseconds, our limit for deploying in-line devices in the core of the network. HTTP response times were also very good, meaning IntruShield could be situated anywhere on a Gigabit network, either internally or at the perimeter.

Once the SYN proxy capability was enabled (this is disabled by default) SYN Flood mitigation was almost total, and 100Mbps of SYN flood traffic had a minimal effect on latency of normal traffic through the device.

Latency increased by only a few microseconds across all packet sizes, although average HTTP response increased from 204ms to 1716ms. No legitimate traffic was blocked during the DOS/DDOS attacks, however.

Section 6: Stability & Reliability

The IntruShield 4010 performed consistently and completely reliably throughout our tests.

Under eight hours of extended attack (comprising millions of exploits mixed with genuine traffic) it continued to block 100 per cent of attack traffic, whilst passing 100 per cent of legitimate traffic.

Exposing the sensor interface to extreme levels of ISIC-generated traffic 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 of the sensor are restricted purely to SSH in order to provide management access.

The extended ISIC attack against the management interface had no effect on the appliance and its ability to detect and block attacks. Alerts were raised to inform the administrator that the management interface was under some form of DOS attack.

The sensor continued to work perfectly throughout and following the ISIC attack, and there were no residual stability problems.