Internet Security Systems Proventia A604

Technical Evaluation

An NSS Group Report

First published March 2005 (Version 1.0)

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

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

TABLE OF CONTENTS

TABLE OF FIGURES

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

The NSS Group is pleased to present the results of its third Gigabit IDS Group Test which includes just two brand new products - a further three products failed our stringent testing requirements and thus do not appear in this report.

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

Overall, the suite contains over **700 individual tests**, many of which are run multiple times, to provide the most thorough and complete evaluation of Network IDS products available anywhere today.

We believe that our test methodology will become the *de facto* standard for testing intrusion detection 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 Detection 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 **Gigabit IDS Group Test (Edition 3)** report can be viewed on-line at *www.nss.co.uk/gigabitids*.

Bob Walder

INTRODUCTION

Whenever a company connects its network to the Internet, it opens up a whole can of worms regarding security. As the network grows, it will play host to numerous bugs and security loop holes of which you have never heard - but you can bet intruders have.

Many organisations are recognising the value of a good security policy to define what is and is not allowed in terms of network and Internet access Then they deploy a number of tools to enforce that security policy – usually in the form of a firewall or two.

Firewalls may be billed as commodity items, but the “shrink wrap” element certainly doesn’t extend to their configuration. A detailed knowledge of what a hacker can do and what should and shouldn’t be allowed through the firewall is required before embarking on the configuration adventure, and a slip of the mouse is all it takes to open up a hole big enough for your average hacker to drive the proverbial bus through. The problem is, a badly configured firewall can be worse than no firewall at all, since it will engender a false sense of security.

To protect an organisation completely, therefore, it is necessary to provide a second line of defence, and in order to achieve this, an entire category of software exists in the form of ***Intrusion Detection Systems (IDS).***

When it comes to computer and network security, there are a number of analogies that can be drawn with the “real world”. Such analogies are particularly useful for answering such questions as “*I already have a firewall, why do I need Intrusion Detection Systems as well?*”.

Depending on how you approach the security of your home, for example, you may opt for high quality locks on your doors and windows. That will help to keep intruders out, and could be thought of as the equivalent of the firewall – perimeter defences. It’s nice to feel secure, but the determined burglar can often find ways around these measures. He can always throw a brick through your back window, for instance, and get in that way – or perhaps you simply forget to lock your door one day.

Once he is inside your home he is free to wreak havoc, perhaps making it obvious he has been there by stealing or wrecking things, or perhaps simply taking copies of any keys he finds so he can come and go later at his leisure. Whatever happens, you don’t want your first knowledge of the break-in to be when you return home to the ransacked contents.

That is why many people install a burglar alarm as well. Should the intruder gain access through the perimeter defences, the burglar alarm alerts you or your neighbours to the break in immediately, and provides an additional deterrent to the would-be thieves.

IDS, therefore, are the equivalent of the burglar alarm. To be used alongside firewalls, they are a recognition of the fact that you can never have a 100 per cent secure system. However, should someone be clever enough to breach your perimeter defences, you want to know about it as soon as possible.

It would also be nice to know what they have been up to while they were inside too.

Intrusion Detection and Vulnerability Assessment are becoming increasingly important as the stakes become higher. In the 1980s and early 1990s, denial-of-service (DoS) attacks were infrequent and not considered serious. Today, successful DoS attacks can shut down e-commerce-based organisations like online stockbrokers and retail sites.

Within the IDS market place there are four broad categories of product:

Host IDS (HIDS)

This category, so often overlooked recently in the face of the more "interesting" technology of Network IDS, is seeing something of a resurgence.

This could be due partly to the fact that high speed switched networks are providing a significant obstacle to effective Network IDS implementation, or it could also be that there is a growing realisation that there is more to IDS than detecting suspicious packets on the wire.

This type of product can itself be split into two main categories, with some host-based systems providing elements of both:

"Traditional" Host IDS (HIDS)

Host IDS (HIDS) products employ an agent that resides on each host to be monitored. The agent scrutinises event/system logs, kernel logs, critical system files and other auditable resources looking for unauthorised changes or suspicious patterns of activity. Whenever anything out of the ordinary is noticed, alerts or SNMP traps are raised automatically.

For instance, a HIDS will monitor the Registry for unauthorised access, kernel logs to detect when inappropriate processes are initiated, or logins to take note of when an attempt is made to access an account with an incorrect password. If a login attempt fails too many times within a short time span the system may conclude that someone is trying to gain access illegally and an alarm can be raised.

Traditional HIDS are very good at detecting insider threats and usually provide extensive damage assessment and data forensics. Bear in mind that the term "insider" does not always refer purely to your own employees. It is possible, for example, for an attacker to gain access to internal systems via a legitimate user name and account combination without having to run any exploit that is detectable by a Network IDS product – perhaps gaining access via *social engineering*. At that point, the attacker would have all the rights and privileges associated with that user, and is much harder to detect.

Disadvantages of the HIDS approach are the need for agent deployment on key systems, and the requirement for close attention to audit policy. They can often be the most difficult of all Intrusion Detection Systems to configure.

File Integrity Assessment (FIA)

File Integrity Assessment (FIA) products monitor the state of system and application files, or the Registry.

They do this by making an initial pass of a clean system and storing a condensed “snapshot” of how that system should look, usually in the form of cryptographic “hashes” of the monitored objects. Once this has been done, it is impossible to tamper with either the original objects or the hash values without invalidating the checksum files. At regular intervals, the FIA product makes a fresh pass, recalculating the checksum values and comparing them against those stored previously.

Thus if an intruder - or Trojan Horse - does manage to gain access to the system and make changes to key files, the FIA product will detect this and raise an alert. This makes FIA the perfect technology for assessing the true extent of the damage inflicted by a successful attack.

The downside, of course, is that because the scans are periodic rather than real time, its strength is in forensic analysis after an attack has been perpetrated, and thus it is of little use where real-time alerting is required.

Network IDS (NIDS)

The *Network IDS* (NIDS) monitors traffic on the wire in real time, examining packets in detail in order to detect patterns of misuse – perhaps spotting denial of service attacks or dangerous payload – before the packets reach their destination and do the damage.

They do this by matching one or more packets against a database of known “*attack signatures*”, or performing protocol decodes to detect anomalies, or both. These signature databases are updated regularly by the vendors as new attacks are discovered.

When suspicious activity is noticed, a network based IDS is capable of both raising alerts and terminating the offending connection immediately (as are some host-based IDS). Some will also integrate with your firewall, automatically defining new rules to shut out the attacker in future.

Most of the network-based IDS available to date work in what is known as “*promiscuous mode*”. This means that they examine every packet on the local segment, whether or not those packets are destined for the IDS machine (much like a network monitor, such as Sniffer). Given that they have a lot of work to do in examining every single packet and tracking active sessions, they usually require a dedicated host on which to run due to their heavy use of system resources.

For instance, most attacks are not based on the contents of a single packet, but are made up of several, sometimes sent over a lengthy period of time. This means that the IDS has to store a number of packets in an internal buffer in order to track established sessions and compare *groups* of packets with its attack signature database. This is known as “*maintaining state*”, and allows IDS to compare new packets against its signature database in the context of what has happened previously in a particular session, rather than examining every packet in isolation.

You will also need one Network IDS sensor per segment, since they are unable to see across switches or routers, and some have problems keeping up with heavily loaded Fast Ethernet segments (never mind Gigabit). Clearly, they would also have problems with encrypted traffic.

Network Node IDS (NNIDS)

The Network Node IDS (NNIDS) is a type of "hybrid" IDS agent which overcomes some of the limitations of the network-based IDS.

The NNIDS agent works in a similar manner to the network-based IDS in that it takes network packets and performs protocol analysis and/or compares them against signature database entries. However, this "micro agent" is only concerned with packets targeted at the network node on which it resides. Because it is installed within the protocol stack of the host, it is sometimes referred to as a *Stack-based IDS*.

Rather confusingly, it is also occasionally referred to as "*host-based*", but usually only by those who are looking at the industry purely from a Network IDS viewpoint. For the purposes of this report, Host IDS is concerned with monitoring of log files and behavioural analysis, whereas both Network and Network Node IDS are concerned with TCP analysis – the only difference is that one (NIDS) is running in promiscuous mode whilst the other (NNIDS) is not.

The fact that the NNIDS system is no longer expected to examine every single packet on the wire, however, means that it can be much faster and take less system resources, and this allows it to be installed on existing servers without imposing too much overhead. It also makes it particularly suitable for heavily loaded segments, switched network environments, or VPN implementations with encrypted traffic on the wire – all areas where traditional network-based IDS' have problems.

Obviously it is necessary to install a number of these NNIDS agents – one for every server to be protected – and they will all have to report back to a central console.

Many organisations may opt for a combination of the two – NNIDS on individual servers in switched server farms, and traditional NIDS on less heavily used segments, where a single IDS can protect a large number of hosts.

Intrusion Prevention Systems (IPS)

Most IDS systems tend to be *reactive* rather than *proactive* – that is they often have to wait until something has actually happened before they can raise the alarm.

The *Intrusion Prevention System (IPS)*, however, attempts to be proactive, and is designed to stop intrusions dead, blocking the offending traffic before it does any damage rather than simply raising an alert as, or after, the malicious payload has been delivered.

Of course, the downside with this approach is the potential for introducing a self-inflicted Denial of Service condition. The thorny issue of *false positives* is one that has plagued the IDS industry to date - sometimes it is very difficult to design an attack signature that will alert reliably on every variation of the exploit whilst ensuring that it will not be triggered accidentally by valid traffic. This is bad enough when you are just being pestered by false alarms at your IDS console, but when the IPS system cuts off a potential customer or your CEO from a vital computer system, the consequences can be far more serious.

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.

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)*.

As with a typical firewall, the NIPS has two network interfaces, one designated as *internal* and one as *external*. As packets appear at the internal interface they are passed to the detection engine, at which point the IDS device functions much as any IDS would in determining whether or not the packet being examined poses a threat. However, if it should detect a malicious packet, in addition to raising an alert, it will discard the packet and mark that flow as bad. As the remaining packets that make up that flow arrive at the IPS device, they are discarded immediately.

Legitimate packets are passed through to the internal 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 or packets with obfuscated URLs will be “cleaned up” before being passed to the destination host.

Seems too good to be true, doesn’t it? Why bother with an IDS at all, since the NIPS device does everything an IDS can do as well as provide true prevention capabilities? Well, when something *seems* to good to be true, it usually *is*!

Firstly, there is the aforementioned potential for an in-line device to introduce a Denial of Service condition by accidentally blocking a legitimate packet flow. If the NIPS is at the gateway to your DMZ containing your e-commerce servers, you had better be sure that *none* of your attack signatures are susceptible to false positives.

For this reason, most adopters of this technology would probably run the NIPS with a reduced signature set that contained only those signatures that had a very low or zero propensity for triggering accidentally on benign traffic. They would then run a standard IDS product on the protected network behind the NIPS which utilised a more complete signature set in order to deal with the alerts that were raised by traffic not covered by the NIPS itself.

The second potential problem is that given the amount of work this device has to do, it can act as a serious bottleneck when installed at the gateway to your network. Most NIPS vendors have recognised this and have sought to get around it by using custom hardware, populated with expensive FPGAs and ASICs. At the time of writing, therefore, adopters of this technology tend to require deeper pockets than those buying plain old NIDS products.

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 when they detect an attack (a worrying piece of FUD that we have noticed in some IDS marketing literature recently). That does not count as *prevention* because by the time the IDS has detected it, the attack payload may already been delivered. Certainly the initial packet that caused the alert will have reached its target, and if the payload is contained in a single packet…. game over! On high speed networks, it would also be possible to deliver an entire payload spread across many packets before the TCP reset command took effect. Only an in-line device (or Host IPS, of course) is capable of real *prevention*.

Gigabit IDS

Clearly, host-based IDS in their various forms are not (or *should* not be) affected by the speed of the network on which they are installed. Therefore whenever we talk about Gigabit IDS we are, by definition, focussing on Network IDS with a Gigabit capability.

Where life gets difficult for those tasked with evaluating this technology is that different vendors have different ideas about what constitutes Gigabit IDS. Some products will be *true* Gigabit products, capable of pulling traffic off the wire for analysis at speeds of up to 1000Mbps (or beyond). Others are merely appliances that contain a Gigabit network card, whose main aim is to allow them to cope with 100Mbps or multiple 100Mbps segments easily.

There is nothing inherently wrong with the latter approach providing the marketing message is honest and does not describe the product as a *true* Gigabit appliance. As long as all the customer needs is to be able to handle 100-200Mbps with confidence - and the price is right, of course - then this is a perfectly valid tactic.

Even true wire-speed Gigabit appliances will have problems in certain areas if they are assembled from off-the-shelf components. At the time of writing, not even the best Gigabit network cards on the market are capable of pulling almost 1.5 million packets per second off the wire, never mind analysing that level of traffic. Thus a Gigabit network loaded with small packets (64 bytes) will cause problems for most Gigabit solutions, and the only way around that for the time being is to move towards custom hardware and ASICs.

Administrators need to be aware of the overall performance limitations of any device when deploying on Gigabit networks. As with most Fast Ethernet networks, the average Gigabit subnet is unlikely to see much more than a fraction of its total available bandwidth in use at any given point in time, and so where only 200-400Mbps is being used, the performance of the Gigabit IDS used to monitor it is less of an issue.

However, one tactic being employed in some organisations is to consolidate multiple 100Mbit segments using a Gigabit switch, and copy all the traffic from each segment to a single mirror, or SPAN (Switched Port ANalyser) port. The Gigabit IDS sensor is then attached to this port to monitor all of the traffic across multiple subnets, thus providing a cost-effective solution to monitoring a number of subnets using a single sensor.

Of course, even if the average utilisation of each subnet is only 40-50Mbps, once you mirror 20 of these you are asking your IDS sensor to monitor getting on for a full Gigabit of network traffic (providing your switch is actually *capable* of mirroring that amount of traffic, of course - an entire topic in itself which is beyond the scope of this report). This is when the performance limitations of some so-called Gigabit devices will begin to manifest themselves.

In some respects, detection performance is the least of the problems facing the administrator tasked with deploying these devices. The problem with any Gigabit IDS product is, by its very nature and capabilities, the amount of alert data it is likely to generate. With 1Gbps of traffic passing through the IDS, the number of alerts could reasonably be expected to be ten times that generated by the typical 100Mbps product. How many members of staff would be needed to process, investigate and resolve that number of alerts? How long before the IDS becomes just another device in the corner that is largely ignored thanks to its insistence on overloading the administrator on a daily basis?

More than ever before in the IDS space, centralised management reporting and forensic analysis is key to the success of the Gigabit IDS appliance. A reduction in false positives (through more accurate protocol decodes and signatures) and global pre-filtering of “safe” alerts that can be ignored are both essential. After that comes the ability to consolidate alerts from multiple sensors to a single management console. Far from increasing the load for a single administrator, this consolidation should help in identifying where potential break-ins are disguised by multiple exploits run over multiple subnets and over a long period of time.

Some management solutions will leave the administrator to determine the connection between exploits, providing the tools to “slice and dice” the data in a myriad of different ways in order to achieve this, whilst others will attempt to perform the correlation in an automated fashion (with varying degrees of success). Either way, the ability to create high-level reports of activity over a period of time, reshuffle and resort alerts in different ways, and then drill down to discover the exact trigger that caused the alert in each case in an efficient manner is now essential.

Without effective management capabilities, alert handling, and forensic analysis tools, the Gigabit IDS is just another lump of iron sitting in the machine room equipment rack.

ISS PROVENTIA A604

Executive Summary

Proventia A Series intrusion detection appliances are designed to detect malicious attacks as they enter the network, including denial of service, intrusions and malicious code, backdoors and hybrid threats like MS Blaster, SQL Slammer, Nimda and Code Red.

Built upon the Proventia code base, the A604 appliance features a pre-installed network agent on a standard Intel platform. The device features two 10/100/1000Mbps ports for management and for sending TCP Resets, and two 100/1000Mbps copper ports for monitoring up to two segments.

The Proventia A604 is designed to support up to 600Mbps bandwidth and offers outstanding performance, easily matching its rated throughput. We also found the A604 to be to be very stable and reliable, coping with our extensive reliability tests with ease and without succumbing to common evasion techniques.

The SiteProtector management system has been well designed to handle management and configuration of large numbers of IPS and IDS sensors across the enterprise. Alert handling is powerful and flexible, especially when combined with the optional SiteProtector SecurityFusion module.

Architecture

The Internet Security Systems appliance-based IPS offering consists of the following components:

Intrusion Detection Appliance

The Proventia A604 appliance is a 1U rack mount server chassis based on a standard Intel platform. Details of the processor and memory configuration are not available, but they are designed to accommodate the rated bandwidth of the appliance (600Mbps in this case).

The device includes redundant internal cooling fans (not hot-swappable), a single power supply and a single disk for data storage. There are two built-in copper 10/100/1000Mbps ports on the rear panel, one used as the management interface, and the other dedicated to sending TCP Reset/Kills. Two more copper 10/100/1000Mbps ports (on a dual-port card) are provided for monitoring two separate segments if required.

The Proventia A604 uses the Linux-based Proventia detection engine together with a custom driver specially built for Proventia hardware. The system is hardened and completely locked down so that the only way it can be accessed is from SiteProtector or SSH (or directly at the appliance via keyboard and monitor). The older Workgroup Manager console is no longer supported with Proventia.

The system is designed to provide all initial configuration functions - such as IP address assignment, management console assignment, and password change - via a simple text-based local interface to which the administrator is restricted following login.

It is also possible to configure network settings for the management interface, set date and time, admin password, TCP Reset/Kill parameters, and reboot or shut down the appliance via this menu. All other day-to-day administrative tasks, such as policy changes, can be accomplished via the SiteProtector manager installed nearby or at a remote location.

Proventia Network Agent

The Proventia Linux-based sensor software (which ISS refers to as an *Agent*) runs on the Proventia hardware in order to monitor a network segment, analyse the traffic flow and look for intrusions and signs of network abuse.

When an intrusion is detected, Proventia can respond in a number of ways, including:

- *Recording the date, time, source, and target of the event*
- *Recording the content of the event*
- *Notifying the network administrator*
- *Terminating the session using TCP Reset/Kills*

Proventia understands over 100 network and application layer protocols such as HTTP, FTP, SMTP, SNMP, RPC, SMB, NetBIOS as well as many Trojan communication protocols. Over 150 protocols are understood, if not completely decoded.

Proventia's protocol analysis capabilities do not rely on merely checking for textbook (RFC) compliance since anomalies are difficult (and sometimes impossible) to define due to RFC ambiguities and differences in interpretation of the standards. Instead, Proventia focuses on analysing protocols to recognise context and then employs pattern matching techniques where appropriate to positively identify malicious content.

In addition, Proventia can also perform full IP packet de-fragmentation, as well as TCP and HTTP session reassembly (tracking over 1 million simultaneous sessions out of the box), and is immune to well-known evasion techniques such as polymorphic shell code mutation, Unicode URL encoding, packet overlapping and RPC record fragging.

Event consolidation logic helps to reduce the total number of unique events that the sensor generates by logically combining large numbers of identical events into a single alert.

Proventia listens to both the transmitting and receiving data streams which allows it to capture server responses for many attacks, such as HTTP, FTP, RPC and DNS events. The ability to report return codes and other server-side responses to the alert console helps guide security operators to quickly recognise the outcome of an attack, and is thus an important tool for prioritising incident response.

Proventia has very strong signature coverage. In addition to the Proventia protocol analysis module, now referred to as the *Protection Engine*, Proventia, like RealSecure, has the ability to import most of the published rules from Snort, via the aptly named *Trons* module.

The aim of this is not to replace Snort, nor is it to enable sites to include the entire Snort signature set (which would seriously impact the performance of Proventia), but to allow users to write their own custom signatures using a relatively simple rules definition language that is familiar to many in the security industry.

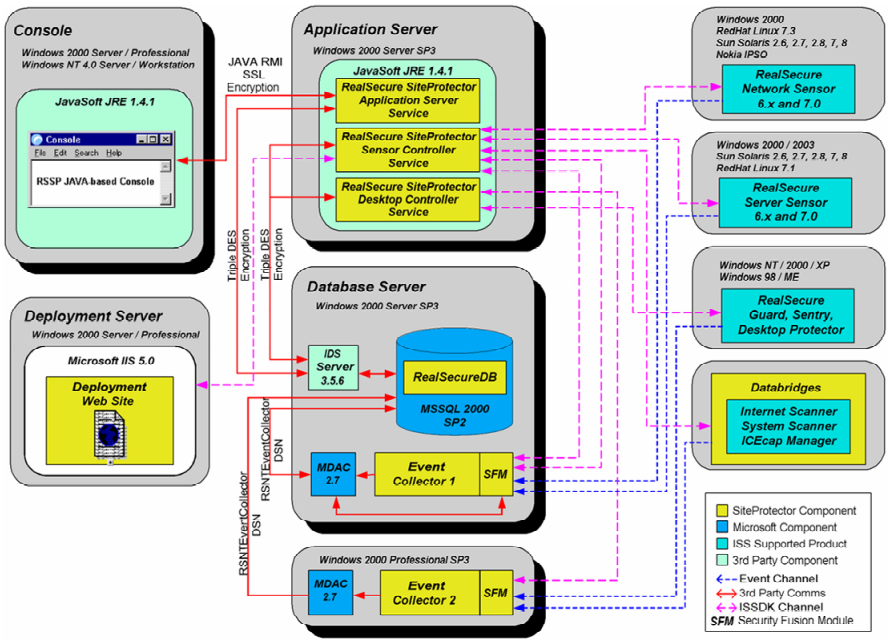


Figure 1 - Proventia: SiteProtector architecture

SiteProtector

SiteProtector provides a central console geared towards large scale enterprise management and event correlation across multiple network, server, desktop, and assessment agents.

Features include:

- *A scalable three-tiered management architecture*
- *Centralised command, control, and event management of network, server, and desktop sensors*
- *Centralised database-driven security analysis*
- *Simplified sensor deployment*
- *Remote, secure, roles-based user interface*
- *Logical "asset-centric" view of security data*
- *Group command and control of sensors*
- *Group-oriented data analysis*
- *Internet Scanner and System Scanner product support*
- *Third-party product integration*
- *"SecurityFusion" real-time event correlation and attack verification (option)*
- *Management reporting module (option)*
- *Automation of security update process*

Deployment Manager

The *Deployment Manager* can be used to install all of the SiteProtector components from a centralised computer anywhere on the network. Once the Deployment Manager has been installed on one PC, the CD is no longer required to install either SiteProtector itself, or any of the other components of a Proventia system. Instead, a simple Web-based installation capability is provided which can be accessed from any host on the network.

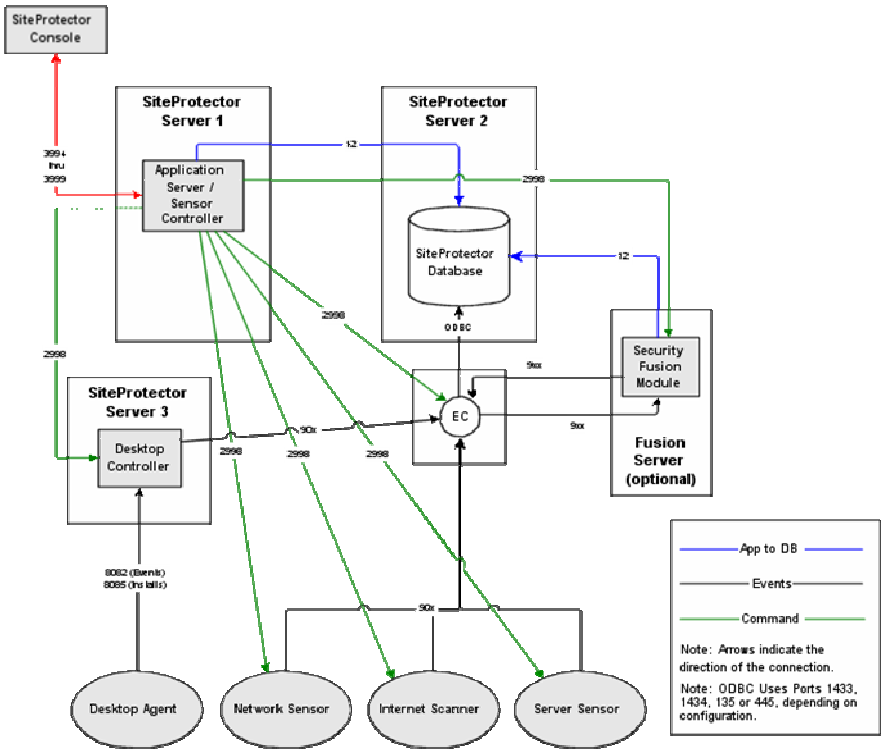


Figure 2 - Proventia: SiteProtector scales across multiple sites

The Deployment Manager can be used to install the Basic or Custom SiteProtector installation, in addition to the following components:

- ***SiteProtector components -*** *Site database, Application Server and Sensor Controller, Event Collector, SiteProtector Console, Desktop Controller*
- ***Add-on components -*** *SiteProtector SecurityFusion module, Internet Scanner Databridge (which is no longer necessary when using Internet Scanner 7.0 for distributed scanning with SiteProtector), System Scanner Databridge*
- ***Components that are managed by SiteProtector -*** *Internet Scanner, RealSecure Network 10/100 and Gigabit, RealSecure Server, Proventia appliances, RealSecure Desktop*

Application Server

The *Application Server* enables communication between the SiteProtector Console and the Proventia Site database.

Sensor Controller

The *Sensor Controller* sends commands to the sensors, such as the command to start or stop collecting events. The Sensor Controller and the Application Server are installed on the same host.

Proventia Site Database

The *Proventia Site Database* stores the data that the Event Collector collects from sensors. This is based on Microsoft SQL Server for large deployments, or MSDE for smaller installations.

Event Collector

The *Event Collector* pulls data from sensors and stores the data in the database. Typically only one Event Collector is installed on a SiteProtector Site, but up to five Event Collectors can be installed per Site when required for performance reasons. The latest release adds failover capability between Event Collectors, such that if one fails another will assume its role.

SiteProtector SecurityFusion Module

The *SiteProtector SecurityFusion* module is an optional (extra-cost) module that correlates data from multiple sources, sources, including Proventia, RealSecure Network and Server agents, and Internet Scanner or System Scanner instances. This automated correlation escalates critical attacks and reduces false alarms.

SiteProtector Console

The *Console* is the graphical user interface (GUI) for the SiteProtector installation. With the Console, the administrator can perform a variety of activities, such as monitoring events and scheduling scans. The specific tasks that can be performed using the SiteProtector Console depends on role assigned to the administrator.

The fully Java-based Site Protector Console included with earlier releases has now been replaced by a Java plug-in to Internet Explorer, providing an increase in performance. Deployment is also easier, since it is no longer necessary to download the Console from the Deployment Manager.

One of the biggest advantages of the Java-based approach rather than the C++ code of the old Workgroup Manager is the ability to spawn multiple windows within the same Console, each performing a different task. So while that large report is loading, it is still possible to be defining and deploying a new policy.

Performance

The aim of this section is to verify that the sensor is capable of detecting and logging 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 sensor can sustain before it begins to drop packets/miss alerts.

The Proventia A604 was tested up to 600Mbps, the rated speed of the appliance. Performance at all levels of our load tests was impeccable, with 100 per cent of all attacks being detected under all load conditions. We would thus have no hesitation in rating the Proventia A604 as a true 600Mbps IDS device as claimed by ISS.

The Proventia A604 performed consistently and mostly reliably throughout our tests. Exposing the sensor interface to an extended run of ISIC-generated traffic had no adverse effect, and the device continued to detect 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, reporting to a single SiteProtector server. We used a modified version of the default *Attack Detector* policy, which had **all** attack signatures enabled (apart from port probes) as well as some key audit-only signatures.

Signature recognition was excellent out of the box (99 per cent), and was increased to 100 per cent after the application of a signature pack update which was provided to us in just 24 hours.

We noted a minimum of “noise” in this release, with very few test cases raising multiple alerts for a single exploit. All our “false negative” (modified exploit) cases were detected correctly, demonstrating that the Proventia signatures are designed to detect the underlying vulnerability rather than a specific exploit.

Resistance to known evasion techniques was excellent, with Proventia being one of the few products to collect a clean sheet across the board in our evasion tests. *Fragroute, Whisker, ADMmutate, running exploits on non-standard ports* and even *RPC record fragging* all failed to trick Proventia into ignoring valid attacks.

Note that not only were the fragmented and obfuscated attacks detected successfully, but every one of them was decoded accurately as well. This is the level of performance to which we would like to see all IDS products aspire.

Out of the box, the Proventia A604 handled over 1 million open connections easily. The sensor also continued to track state on our “half open” exploits beyond its configured maximum open connections, since Proventia is designed to ignore new connections once the limit is exceeded, but does not age out old ones.

If you prefer old connections to be aged out in favour of allowing new ones when the state tables are full, ISS has added a parameter in the latest release which makes this behaviour configurable too.

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

As you would expect from an appliance that is designed to be as close to "plug and play" as possible, installation is very straightforward. Initial configuration is performed at the appliance, the simple configuration menu being presented automatically when logging in to the admin account. The admin interface on the Proventia A604 is designed to disallow any OS modifications, and the Root account should never be used - ISS will not support any configurations that are not accomplished from the admin account.

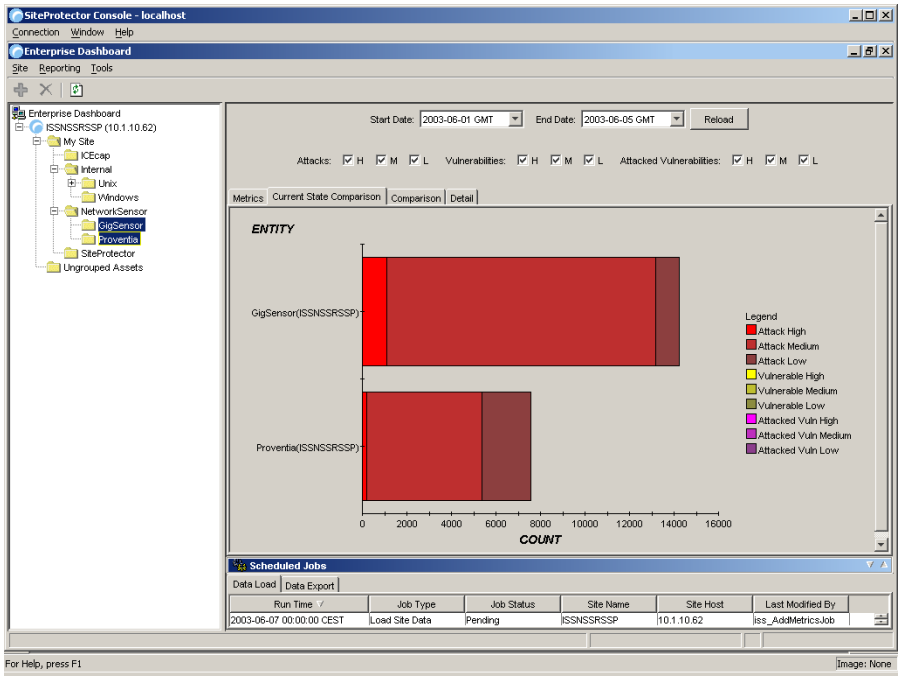


Figure 3 - Proventia: Enterprise Dashboard

The simple, text-based menu enables the administrator to configure network settings, date and time, admin password, allow or disallow SiteProtector access, restart the protection Agent, configure the TCP Reset/Kill parameters, reboot and shutdown the appliance. A recovery CD provides the means to completely re-install the appliance software should that prove necessary (following disk corruption, for example).

Once configured, the Proventia appliance can only be accessed via a serial cable using a communications application such as Hyperterminal, keyboard and monitor, SSH, or SiteProtector.

Documentation is generally excellent for the Proventia range. It is provided as PDF files or hard copy, and includes an *Installation Guide* and *User Guide* for each of the components. The level of detail is good, and we found coverage of all the main features to be reasonably comprehensive and very clear.

The *SiteProtector Strategy Guide* is an extremely useful additional manual which provides best practice guidelines and suggestions for securing a network, offering a guide to deployment and use in a range of organisation sizes and complexities.

Configuration

SiteProtector provides scalable, centralised security management and data analysis capabilities for all Proventia appliances, RealSecure Network, Server, and Desktop agents, and ISS' scanning applications. SiteProtector simplifies Proventia deployments through unified command, control and monitoring, the aim being to reduce security management demands on network traffic, staff or other operational resources.

The first goal of SiteProtector is to provide the administrator with a more logical view of the security boundaries under his control. Thus, rather than view individual sensors by machine name or IP address, they can be grouped together logically into departments or physical sites.

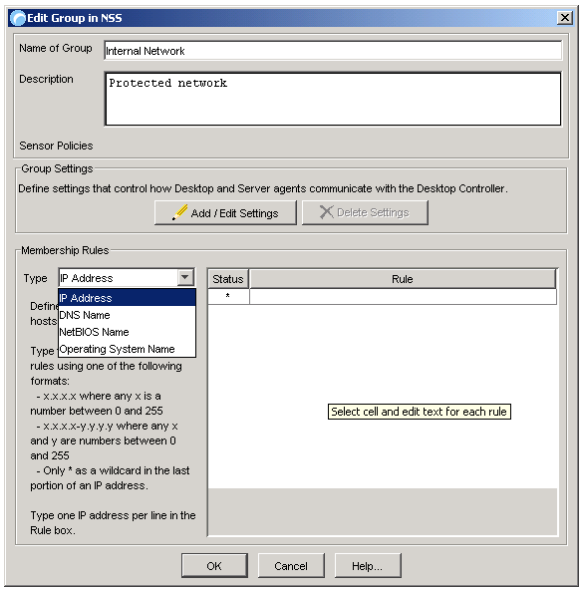


Figure 4 - Proventia: Creating groups in SiteProtector

This makes it much more meaningful when viewing alerts or applying security policies.

Security assets can belong to more than one group for maintenance and reporting purposes, but can be “subscribed” to a single group for policy application, forcing policies to be deployed automatically each time they are amended. It is also possible to override this and apply policies on a per-sensor basis. This has been very well thought out, and provides one of the most flexible policy deployment capabilities we have seen.

Administrators can manage a wide range of sensors - both network and server-based - across the corporate network from a single Console if required. It is also possible, of course, to provide multiple Consoles to different administrators, and each one can be assigned different roles. For example, an administrator may control, configure and maintain the SiteProtector system and its components at the same time that localised security analysts perform command and control over the sensor infrastructure.

However, these analysts remain restricted from configuring the SiteProtector system itself. Operators with viewing privileges cannot perform command and control functions, but may view the aggregated security information contained within a SiteProtector environment.

As with previous versions of the software, users are designated by assigning Windows users to specially created groups, which designate *Administrator*, *Operator* or *Analyst*. Within SiteProtector, these users can then be permitted or denied access to individual resources - groups or sensors - within the Site. Users can also be permitted global access to all Sites with a single login, or be forced to authenticate to each site individually as they drill down when performing analysis.

As new sensors are installed they can be registered manually or automatically (via a scanning operation) with the SiteProtector Console. As new assets (hosts or sensors) are registered, it is possible to auto-group them according to pre-defined rules (such as domain name, IP address range, and so on).

Since default policies can be applied at group level, it is possible to install a sensor, have that sensor register itself, assign it to a group and apply a policy without any direct action being required from the administrator. This is a nice feature.

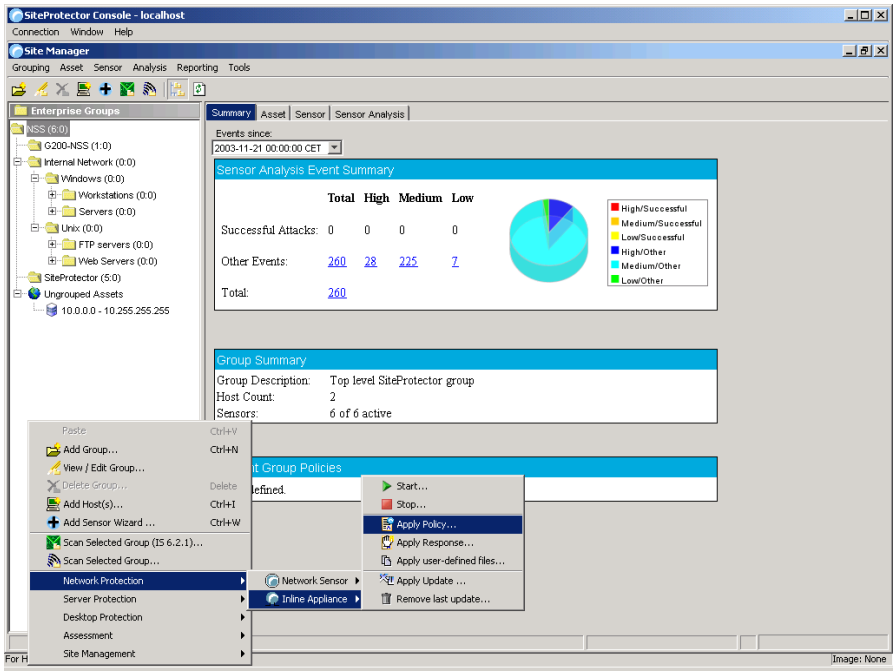


Figure 5 - Proventia: Managing groups within SiteProtector Console

Command and control of a group of sensors can be achieved with a single click by selecting multiple sensors and right-clicking, or by applying management operations at group or site level, in which case all sensors beneath are affected simultaneously.

Context-sensitive menus appear following a right click to provide the option to start and stop sensors, apply policies, apply global responses, apply updates, remove updates, or run vulnerability scans (if Internet Scanner is installed). Whenever a command and control operation is selected, the option is provided to run once or to schedule multiple repeated operations.

This would allow the administrator to have different security policies applied across a range of sensors on weekdays to those applied at weekends, for example. The potential time savings for large-scale deployments are enormous, and the ability to organise assets by multiple nested groups within multiple sites and apply management operations at any level of the hierarchy make the system extremely scalable.

Key database management tasks can also be performed via the Console. Automated backups can be configured in the user interface by right clicking on the database, as well as sophisticated automated purging by event categorisation type. This latter feature enables the administrator to set a time interval for keeping exceptions, incidents and un-categorised events, and to set granular filters on specific events to define exceptions to the automated purging regime.

Policy Management

The most important configuration task, of course, is to assign a security policy, and four default - and fixed - policies are provided with the system (additional policies are available for backward compatibility with previous versions of RealSecure). Different policies are defined with different modes of protection in mind.

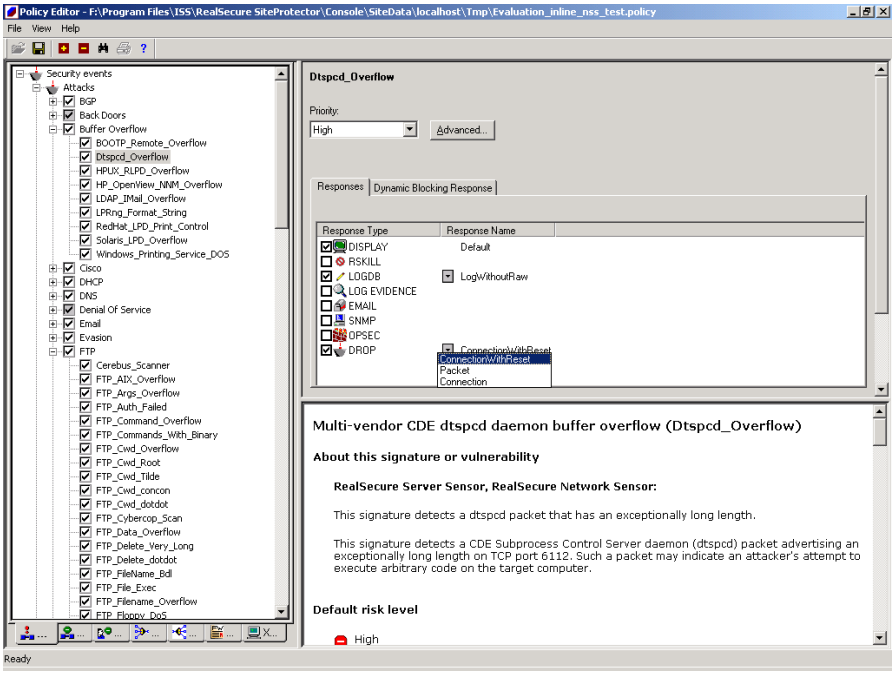


Figure 6 - Proventia: Policy Editor

For example, one has all attacks **and** audits enabled, whilst another has attacks only. There is also an "*Evaluation*" policy which has all attacks enabled (apart from port probes) along with certain key audit signatures. There is no performance impact when all signatures are enabled. Even if a signature is turned off in a policy with Proventia, it is ignored for alerting purposes only, but is still included in the attack detection phase. Thus the Proventia effectively *always* runs with *all* signatures enabled.

The Policy Editor used in SiteProtector is the one remaining legacy of the old Workgroup Manager product, and is due to be replaced throughout the product range later this year by the new - and much slicker - Java-based *Common Policy Editor*.

It is certainly easy enough to create new policies - simply select one of the existing policies and click on "*Derive New Policy*", give it a name, and you are free to create your own policy using that as a template. There are six tabs on the policy definition screen, including:

- *Security Events*
- *Connection Events*
- *User Defined Events*
- *Packet Filters*
- *Event Filters*
- *X-Press Updates*

The *Security Events* tab lists all the available attacks in the signature database, and these can be enabled or disabled by clicking on a check box next to each one. The available signatures are split into two categories - *Attacks* and *Audits*. *Attacks* are those events which are deemed serious enough to warrant an alert (DOS attacks, Trojans, Web exploits, etc) on any network.

Audits are events which may be considered suspicious but may be prone to false positives if enabled on certain networks. For example, one Audit signature monitors all SNMP activity, which is only of use if you do not run SNMP on your network. Other Audit signatures monitor "reconnaissance" events such as FTP SYST, SMTP EHLO, or Bind Version requests.

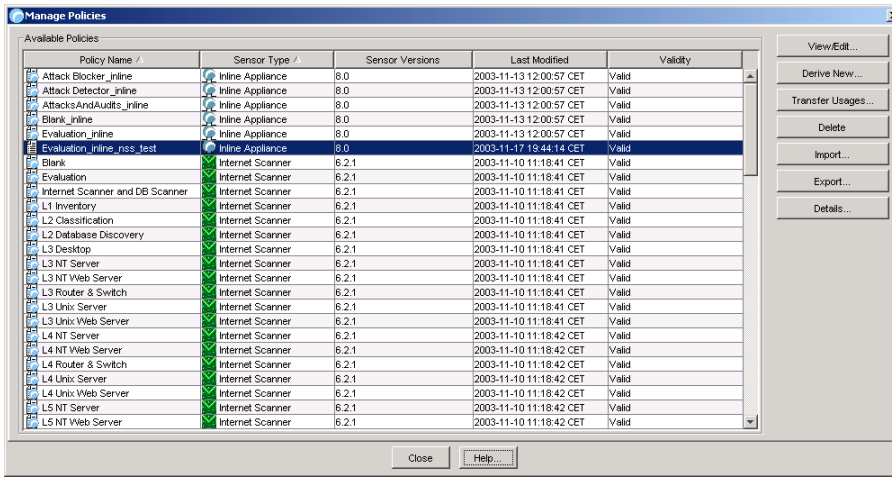


Figure 7 - Proventia: Managing Policies

Most companies would begin with the *Evaluation* policy, which includes all the serious attack signatures and some key audit signatures (this is basically the one we used for our tests). This can then be tuned to remove any false alarms and add in any Audit events which may be useful on a particular network. The *Attacks and Audits* policy would provide the most complete coverage out of the box, but would provide far too many false alarms and superfluous data on most networks.

Selecting an individual attack brings up a detailed description of the attack (including its effect and how to counteract it) and a configuration screen.

There is a wide range of attacks available, and new ones are added at regular intervals through the efforts of the Internet Security Systems X-Force team. The Web-based update procedure is very straightforward and easy to use, and allows the administrator to download new signature files (known as *X-Press Updates* (*XPU)*) to the network and update individual Proventia installations from those, all of which can be accomplished from the Console.

Unfortunately, new signatures are only included on the XPU tab, and not in the main Security Events tab, which makes them very difficult to find unless you know exactly where they are. This problem should be eliminated by the use of the *Search* function, but for some reason this never searches beyond the first occurrence of the search string entered. Policy definition can thus be more difficult than it should be. It would be nice to see the Search function extended to produce search results consisting of multiple signatures, and also allow searching on additional criteria.

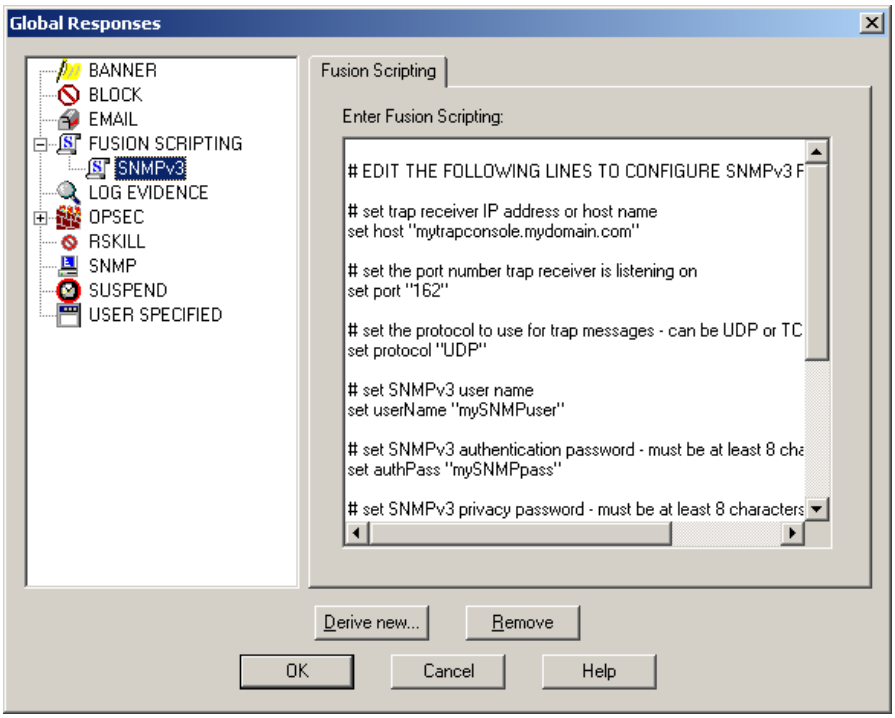


Figure 8 - Proventia: Configuring Global Responses

The attack priority sets the priority flag which can be used for filtering and reporting, and this can be escalated automatically by the Fusion module if it determines that an attack was successful. For each event there are also a number of responses available, including:

- *Notify Console*
- *Log to database*
- *Log raw data*
- *Log evidence*
- *Send e-mail notification*
- *Kill connection*
- *View session*
- *Re-configure firewall (OPSEC)*
- *Send an SNMP trap*

Kill connection resets the IP connection to terminate the attack immediately, whilst the OPSEC option allows Proventia to automatically reconfigure any OPSEC-compliant firewall to prevent further attacks.

Response settings become the defaults for individual signatures, but these can be overridden in bulk by applying different response settings at sensor, Group or Site level.

There is also an *Advanced* properties screen that once provided the means to edit any additional parameters that might apply to a specific attack signature, as well as to define how multiple events should be handled and propagated from sensor to Console.

This has changed following the integration of the BlackICE software, and both the Event propagation control - the means by which rapidly occurring attacks can be consolidated into a single alert to prevent network floods - and tuning parameters for individual signatures are now made available via a set of parameters that are applied at the sensor level.

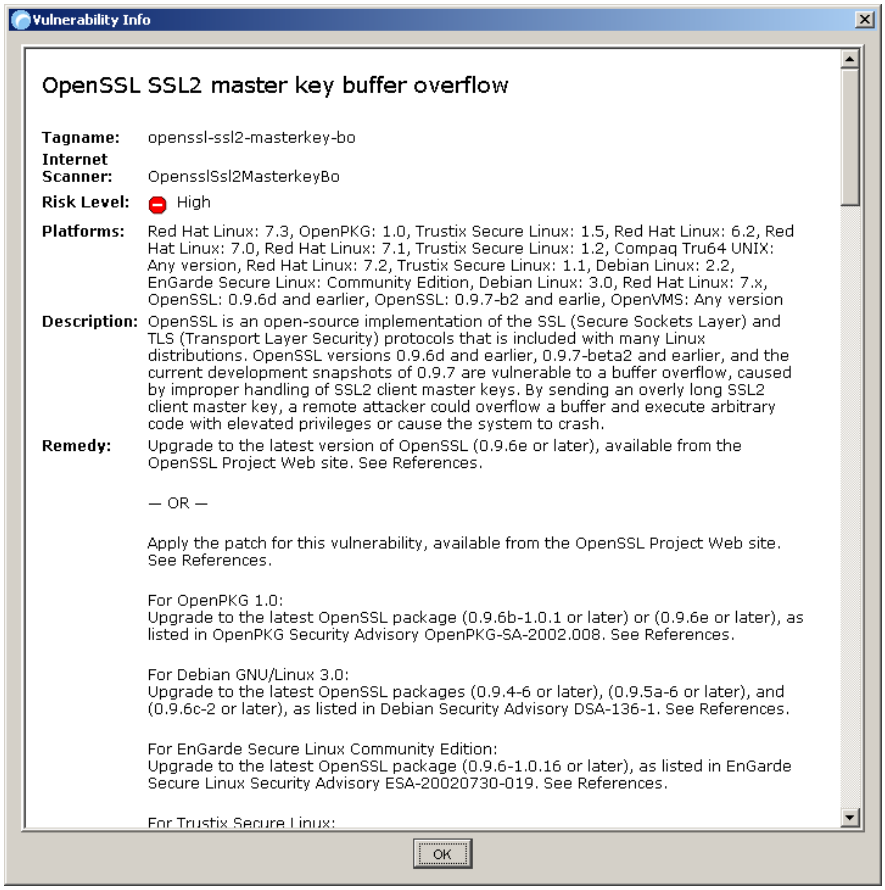


Figure 9 - Proventia: Vulnerability information stored against each signature

Event coalescing worked extremely well in our tests, with each alert generated containing a repeat count to show how many attacks of that type were actually detected. This count remains accurate until the sensor is under extreme loads, at which point a “pre-coalescer” cuts in to drop a percentage of identical packets before they are passed to the parsing engine.

Proventia can be used to monitor more than just security problems by using the *Connection Events*. These are generic events such as HTTP, FTP or SMTP activities, and can be filtered by source or destinations address, source or destination port, or protocol. One possible example of how this could be used would be if company policy forbids the use of Telnet. Proventia could be configured to monitor for TCP port 23 and log the source and destination addresses of the perpetrators, perhaps blocking the process completely too.

User-defined signatures allow the administrator to apply regular expression string matching on certain fields within a packet (login name, URL data, e-mail subject, etc.) to determine suspicious content. This could be used as a "quick and dirty" virus checker by creating a signature that raises an alert every time it detects a packet with a specific string in the e-mail subject field, or to monitor and alert on attempts to login as Root.

User-defined signatures containing regular expressions *can* be a performance drain. However, it was not necessary to turn off RegExp parsing of custom signatures in order to achieve 1Gbps detection rates. Note that this is the limit of signature customisation via the Console interface – the built in signatures remain virtually untouchable. However, Proventia does include the *Trons* module which provides the means to add completely custom rules using the Snort rules definition syntax.

Although neither TCP reassembly nor any of the standard Snort plug-ins and pre-processors are supported, Trons does allow the administrator to add brand new signatures using a familiar "language" that is reasonably easy to learn and simple to use. And, of course, there is an entire library of signatures ready made on the Snort Web site, though Internet Security Systems does not recommend you add them all.

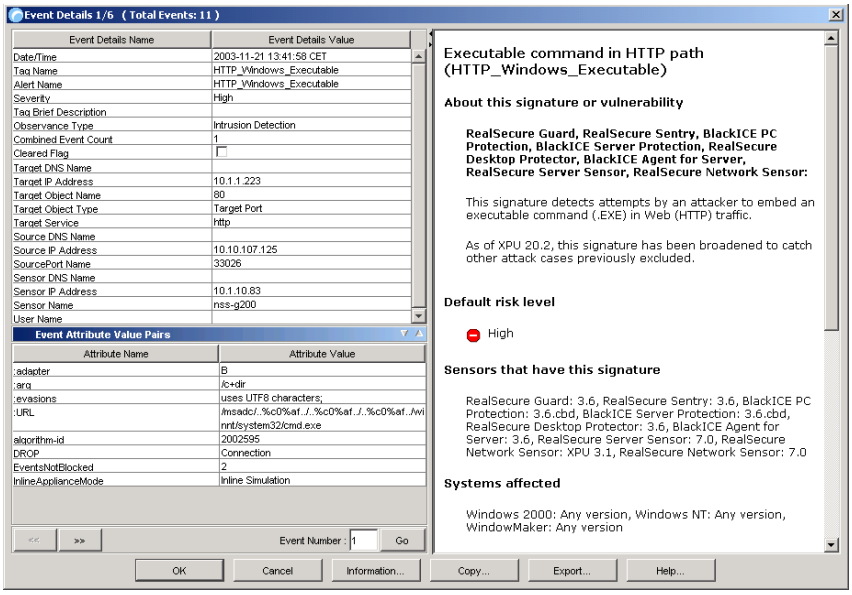


Figure 10 - Proventia: Viewing Event details

Apart from a significant overlap in functionality between the Snort and built-in signatures, there would be a heavy performance impact. Because the Trons module is currently completely separate from the Proventia PAM (this will not always be the case in future releases), the signature parsing path is not exactly optimised, and thus Trons signatures should be kept to a minimum. The Trons module is also disabled by default.

We found it very straightforward to add Snort signatures to our Proventia sensor, and the resulting alerts are displayed in the SiteProtector Console alongside all the built-in alerts, but including Snort-specific data in the Event Inspector window, such as the Snort SID, external reference, version number, and so on.

Finally, if there are genuine events which are raising false alarms via Proventia, the *Packet Filter* tab allows the administrator to define which packets should be ignored for alerting purposes, based on protocol, port or IP address. All packets matching one of the defined packet filters are rejected before being processed by the parsing engine.

Where it is still required to disregard specific events that have been detected by the sensor - perhaps because they are known to be false positives when coming from a particular host - then the *Event Filters* can be used. Here, the administrator can specify an event name, together with source and destination IP address and port, and any event detected by the sensor matching these criteria is simply ignored.

Once policies have been defined they are applied to individual sensors, Groups or Sites via the SiteProtector Console. Multiple policies can be defined for different functions (i.e. DMZ or internal network) and applied to a range of sensors via the asset hierarchy. Policy application can be performed as a one-off process or scheduled for regular runs, thus allowing administrators to apply different policies at different times (days and nights, weekdays and weekends, and so on).

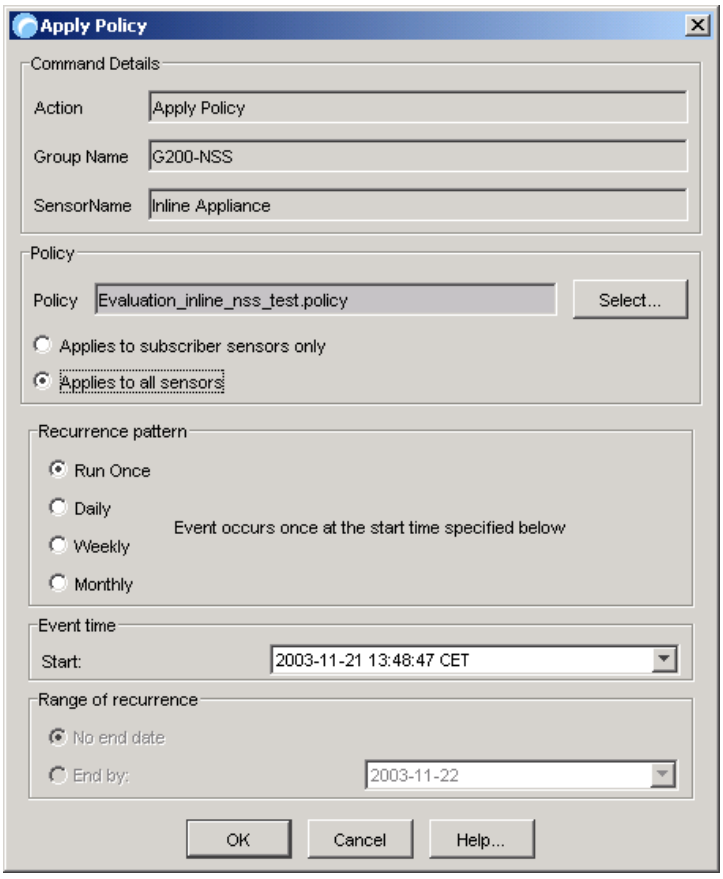


Figure 11 - Proventia: Applying Policies

If the administrator makes a change to an existing policy, SiteProtector will determine which sensors are using that policy and provide a list to the administrator allowing him to deploy the policy to all sensors in one go, or just a selection of them (by deselecting checkboxes).

The existing Policy Editor in SiteProtector 2.0 is actually one of the few remaining legacies from the old Workgroup Manager Console. This will be replaced in the near future by a new - and much more feature-rich and intuitive - Java-based *Common Policy Editor* that will provide granular policy control and a single interface for all types of security policies to be applied via SiteProtector (i.e. not just for Network IPS/IDS products, but also for vulnerability assessment, host-based protection, and so on).

Alert Handling

Once a sensor is up and running with a policy applied, alerts are logged locally on the sensor (if required) and then forwarded to the designated Event Collector, from where they are stored in the database. At user-defined intervals, the SiteProtector Console retrieves new events and displays them in the *Analysis* pane in a “spreadsheet” format.

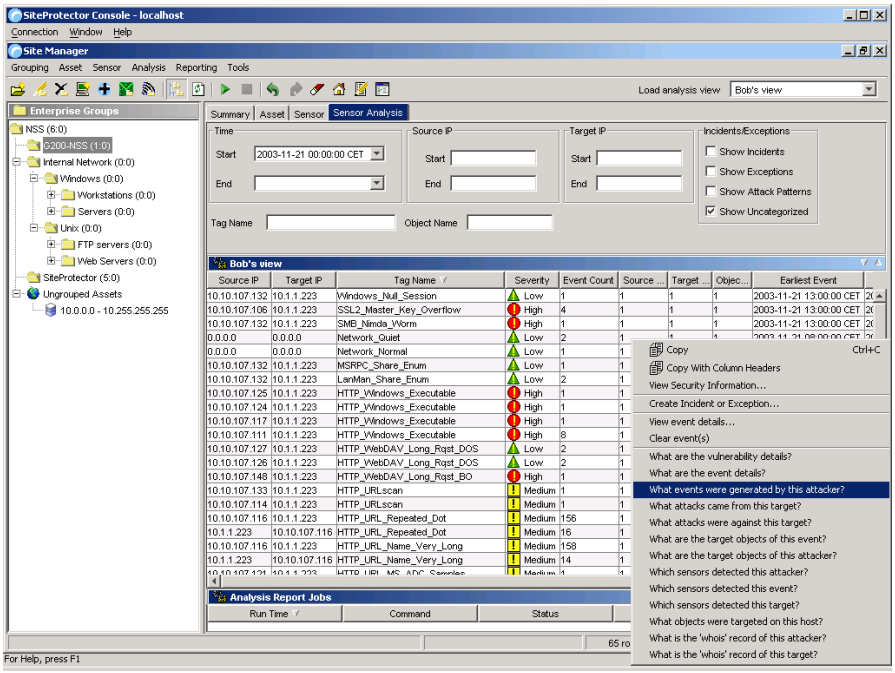


Figure 12 - Proventia: Viewing security alerts

Each row of the spreadsheet displays the complete alert details, including date and time, source and destination IP address, source and destination port, severity, status, URL/command that cause the alert, and so on. Right clicking on an alert provides the ability to view all alert details in a single window, which makes it more readable. It is also possible to call up the detailed X-Force information on that alert,

Although the ability to view detailed packet contents is missing, a configurable parameter within the sensor enables full packet logging, the results of which can be viewed via a third party tool such as Ethereal. There is no link from any particular event to a specific packet, or group of packets, within the log file, however.

Of greater value is the ability to provide extended context information as part of the event. This information is collected for every stateful session being tracked by the sensor, and is collated and made available to the Event Collector whenever a new alert is raised. This information includes data specific to the actual alert - such as the offending URL or buffer contents in the case of an HTTP overflow - as well as pertinent data collected from packets leading up to the exploit - such as user name and password used to log in to the server in the case of an FTP session. In most cases this is of much more use than detailed packet contents, since the data leading up to the packet which triggered the exploit is often of equal importance to the contents of the trigger packet itself.

A number of other options on the right-click menu guide the administrator through the process of "drilling down" through the data to get to the most important and relevant information. This is done by providing a number of plain English "questions" such as "*What are the event details?*", "*What events were generated by this attacker?*" and "*What attacks were against this target?*". A number of filter options are provided along the top of the screen, allowing the administrator to home in on the data that is of interest by specifying source and target IP addresses (or ranges), ports, date and time stamps, and so on.

One extremely useful feature is the ability to create "baselines". At any given point in time, the current analysis view can be "frozen", which maintains the totals and counts (such as event count, target count, and so on) on screen. Any increases or decreases in those totals are then shown in red as plus or minus figures from the baseline, making it easy to spot trends during an investigation.

The baseline enables the administrator to determine, at a glance, how events have changed in an analysis view. If, for example, he notes that one IP address or tag name is associated with an unusually high *increase* in the number of events, he can investigate it to determine whether it represents a threat.

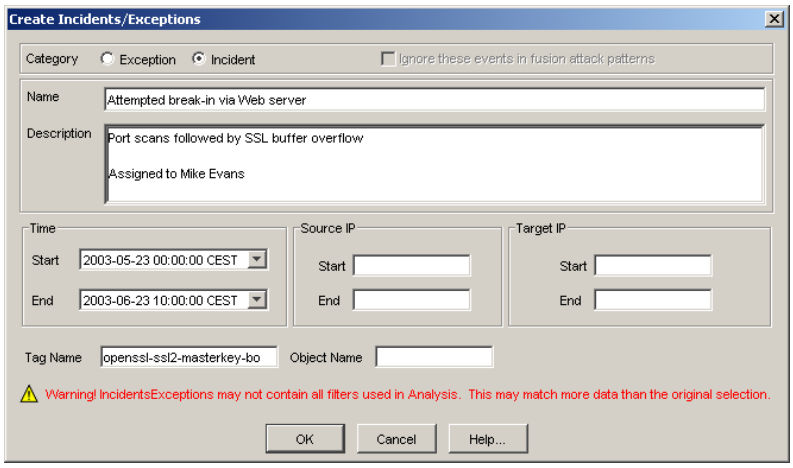


Figure 13 - Proventia: Creating Incidents and Exceptions

Different analysis views can be loaded directly from a drop-down menu, each one providing a different data layout (a different "report format" if you will), and these are almost infinitely customisable via the ability to define column layouts and filters to be applied to the underlying data. Once the administrator has fine-tuned the analysis to his satisfaction it can be saved for subsequent recall.

Events that are deemed unimportant can be designated as "*Exceptions*" and they will subsequently not appear in analysis views. This does not prevent them from being detected in the first place, merely from being displayed for analysis - it would be nice if there were a right-click option to enable the signature that caused the alert to be disabled in the policy.

Should the administrator determine that a group of events are related (perhaps a port scan, followed by a buffer overflow, followed by attacks launched from the compromised machine) they can be groups together as an "*Incident*". These events are then removed from the normal analysis display and are shown only in the Incident analysis views. This allows the administrator successively to reduce the data displayed, resulting in the ability to "find the needle in the haystack".

Incidents are tracked as a unit from that point on, and the incident can be annotated with actions taken to resolve it. This is an extremely useful feature, and would be even more useful if it were possible to flag each incident with a status (pending, resolved, under investigation, etc) and an owner.

SiteProtector's modular format enables plug-in enhancements for a wide range of security management needs, and the *SecurityFusion* module is the first plug-in module for SiteProtector. This extra-cost module uses data correlation and analysis to rapidly and automatically derive the likelihood of a successful attack from aggregated vulnerability assessment information. Visual cues in the SiteProtector Console indicate attacks with a high probability of success, with automatic escalation criteria for critical security events.

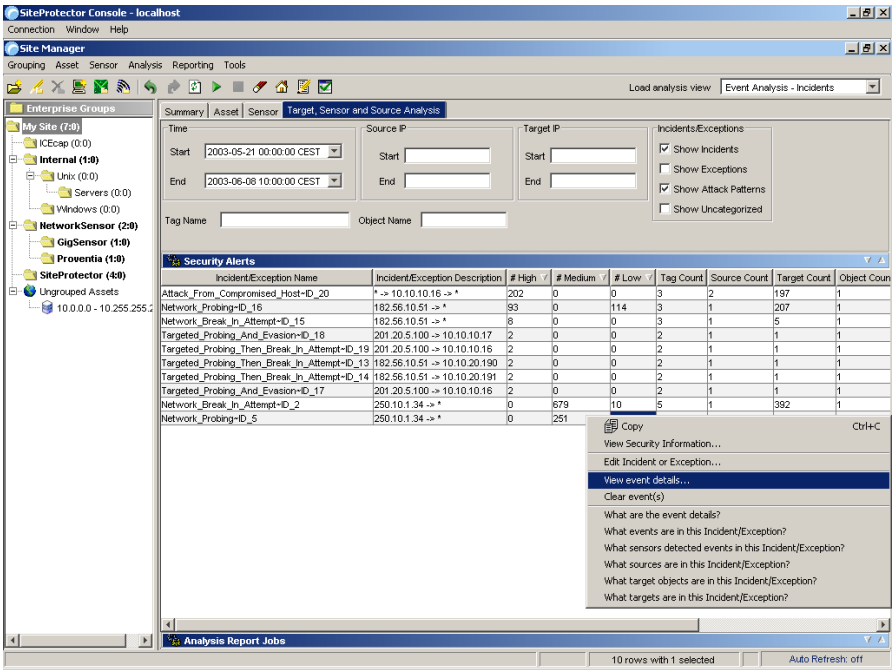
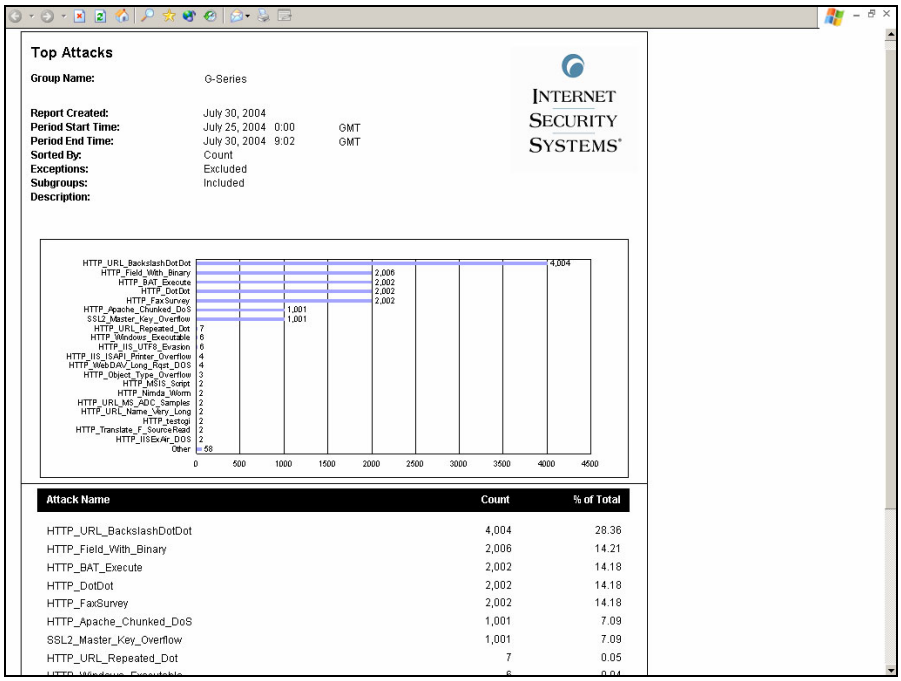


Figure 14 - Proventia: Investigating Incidents created by the SecurityFusion module

For example, the module can escalate important events by generating additional responses outside the Console (such as email or SMTP). Alternatively, it can de-emphasise less important events by reducing alert priority or by selectively preventing an event from being displayed or logged. All escalation and de-escalation options are fully customisable.

During testing, we noted that a specific alert - a DNS Zone Transfer - was escalated from the normal event priority of *medium* to *high* because Internet Scanner had previously determined that the particular host against which the transfer was made was potentially vulnerable to spurious zone transfer requests. On the other hand, it would be possible to "downgrade" an alert from high to medium or low if the attempted attack - say an FTP exploit - was against a host with no vulnerable FTP service running.



Top Attacks

Group Name: G-Series

Report Created: July 30, 2004
Period Start Time: July 25, 2004 0:00 GMT
Period End Time: July 30, 2004 9:02 GMT
Sorted By: Count
Exceptions: Excluded
Subgroups: Included
Description:

INTERNET SECURITY SYSTEMS

Attack Name	Count	% of Total
HTTP_URL_BackslashDotDot	4,004	28.36
HTTP_Field_With_Binary	2,006	14.21
HTTP_BAT_Execute	2,002	14.18
HTTP_DotDot	2,002	14.18
HTTP_FaxSurvey	2,002	14.18
HTTP_Apache_Chunked_DoS	1,001	7.09
SSL2_Master_Key_Overflow	1,001	7.09
HTTP_URL_Repeated_Dot	7	0.05

Figure 15 - Proventia: Typical report

This approach can help to reduce false alarms, and can certainly reduce the load on the administrator since it would be possible to record all suspicious events for trend reporting and forensic analysis, whilst only alerting on events where there is a real chance of an exploit targeting a vulnerable host.

The potential downside, of course, is the necessity to run Internet Scanner at sufficiently regular intervals to make the data correlation effective, but SiteProtector makes it easy to schedule Scanner runs in order to achieve this. In addition to the escalation and downgrading of alerts, Fusion also records its assessment of the alert (failure, likely to have succeeded, etc.) in the alert details, making it available in analysis views and reports.

The most valuable feature of Fusion, however, is its correlation capability - the ability to analyse and group alerts into Incidents. If Fusion detects a pattern of events similar to the one we mentioned earlier - a port scan to a particular host, followed by a potentially successful exploit, followed by further port scans and exploits launched **from** that host (indicating that it had probably been compromised) - then it would group all of these alerts together into a single Incident.

Because alerts within Incidents are removed from the normal analysis review, the result is a much smaller number if unclassified events to deal with.

And by focussing on the Incident analysis first, of course, the administrator can be sure of spending his time dealing with genuine problems. Naturally, when operating inline it still remains within the administrators control whether to block the individual events or to alert only.

The alert-handling capabilities of SiteProtector 2.0 are a huge improvement over previous RealSecure management consoles, and this is one of the better IDS consoles we have seen in our labs.

Reporting and Analysis

ISS has provided an almost infinitely customisable analysis tool in the shape of the *Sensor Analysis* tab that we discussed in the *Alert Handling* section.

A number of very useful basic views are provided out of the box, including *Attacker, Details, Event name, Incidents, Sensor,* and *Target.* However, the real power of the system lies in the fact that it is possible to create a limitless supply of customised views by specifying your own column layouts and filters. These custom views can be saved for later recall, and can also be saved as reports in PDF, HTML or CSV format.

When saving HTML files, an index is automatically created of all reports saved, making it easy to publish management reports to a Web site if required. All reports can also be scheduled to run at regular intervals, and query performance has been improved significantly in the latest release.

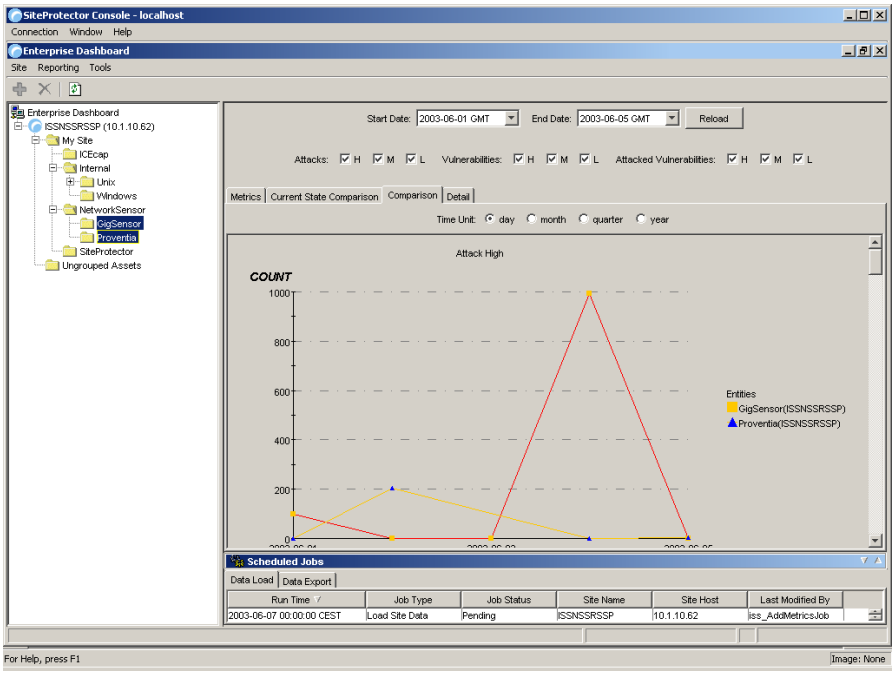


Figure 16 - Proventia: Sensor comparison in the Enterprise Dashboard

In addition to the analysis views, SiteProtector also provides the Enterprise Dashboard. This allows the administrator to:

- *View a high-level graphical summary of alert activity for a site (this is the default view in the new release)*
- *View metrics and graphs for an enterprise or a site*
- *Export reports*

- *Create groups for specific sensors or assets monitored*
- *Perform high-level command and control functions, such as associating users and groups, which determines the specific sites, groups, and subgroups that users can access*
- *Drill down to examine site data in more detail*

The *Metrics and Trends* pane enables you to view security results for specific assets, sensors, and sites, and to create reports that display trends. The information in the Metrics and Trends pane is based on the filters you select in the Filters pane. This displays information on the following tabs:

- ***Metrics -*** *Volume of events by type of event for the group selected. The administrator can select multiple groups for the display.*
- ***Current State Comparison -*** *Bar chart of event volume by priority and type of event. It is possible to select multiple groups for the display.*
- ***Comparison -*** *Trends for the type of events by category and priority. Trend lines are displayed for each selected group, and it is possible to select multiple groups for the display.*
- ***Detail -*** *Trended stacked chart showing the volume of events. The administrator can select multiple groups for the display, and data is displayed for the dates chosen in the time filters.*
- ***Configuration****—Information is shown in tabular format about the selected groups. Information includes name of group, its site, and the applicable policy or X-Press Update.*

Even though the detailed packet contents are not stored as part of the Proventia database, it is possible to store detailed packet logs of all traffic seen by the sensor (packet logging) or just packets associated with specific signatures (evidence logging).

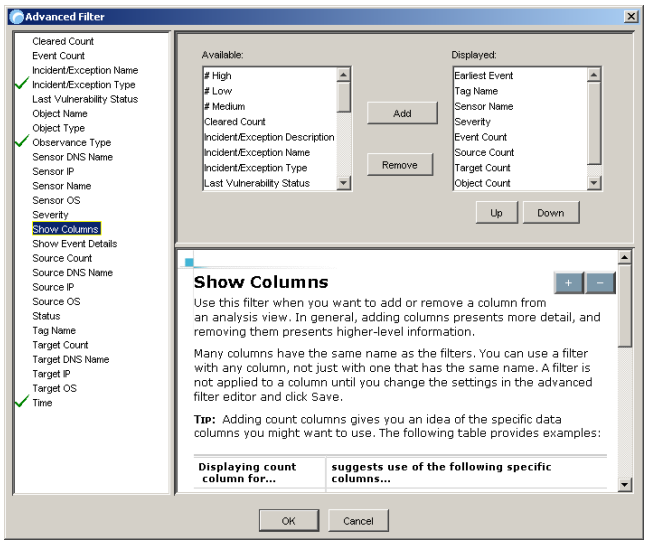


Figure 17 - Proventia: Creating custom filters for reporting

When packet logging is enabled, Proventia records all system traffic into log files, the size of the files and rotation characteristics controlled via the *Advanced Properties* tab for each sensor. It is important to note that packet logging keeps track of **all** system traffic, not just intrusions, which can result in some large files and would obviously have quite an impact on performance.

If it is required to be more selective in what is recorded, evidence files can be used. These are controlled via the global or per-signature settings in the policy files. Whenever a particular attack is detected that is flagged as "*Log Evidence*", Proventia captures network traffic specific to the attack in progress and stores that information in an evidence file.

Both Packet and Evidence Logs are encoded as *Sniffer* or *tcpdump* trace files, and will require a decoding application (which is not included) to view the contents. Unfortunately, these log files must be retrieved manually from each sensor (although they can be retrieved centrally via the Console) and there is nothing to tie the individual alerts in the database to specific evidence files, making forensic analysis more difficult than it should be.

Finally, the latest release sees the introduction of text-based and graphical management reports, provided by an extra-cost option based on the ubiquitous Crystal Reports product. There is not much within this module that will interest the forensic analyst, since these are aimed more at producing higher-level summary reports, providing pre-defined templates *for Top Attacks, Top Attack Targets, Top Attack Source, Attack Incidents, Attack Trends*, and so on. There are also a number of templates that rely on information produced by other modules besides the Network IPS on test here, including numerous assessment (from Internet Scanner), desktop protection, and virus reports.

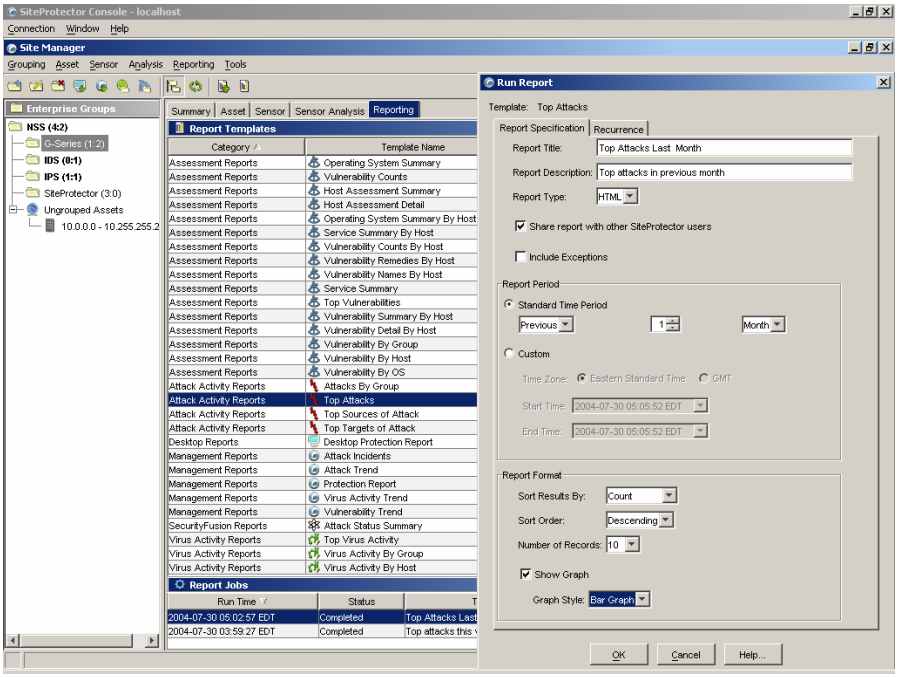


Figure 18 - Proventia: Creating reports

They will certainly be of interest to those who require to monitor trends in the type and severity of exploits detected, or in their effectiveness at handling security incidents, and are thus aimed more at manager-level. This is one area that was omitted from the early Site Protector releases, and this module plugs a gap for those who require more than straight event analysis capabilities.

In selecting a report template, the user is presented with a screen which enables him to name the report, select a report period, and a report format.

The report period settings provide an excellent range of pre-defined date ranges, as well as the ability to specify your own down to the nearest minute - we suspect that most users will opt for the “current week” or “previous month” settings at the click of a mouse, and wish that all vendors would allow date ranges to be specified in this way.

The formatting options are fairly basic, usually limited to selecting a sort order, number of records to be displayed, and whether or not to show a graph. The output of the reports is thus fairly fixed, but it is tidy, well-presented and easy to read.

Verdict

Performance

Clearly a lot of effort has been put into designing the Proventia hardware to cope with the bandwidth claimed (600Mbps on the A604) and in producing custom drivers for the built in network cards. This care allowed Proventia to detect and log 100 per cent of attacks under all of our test conditions. We would have no hesitation in rating the Proventia A604 as a true 600Mbps IDS device, and are confident that it could actually perform well beyond this level on any normal network.

The Proventia A604 also performed consistently and reliably throughout our tests, continuing to detect attack traffic in a consistent manner even when under extended attack. Exposing the sensor interface to an extended run of ISIC-generated traffic had no adverse effect, and the device continued to detect and log all other exploits throughout and following the ISIC attack.

Security Effectiveness

Signature recognition was excellent out of the box (99 per cent), and was increased to 100 per cent after the application of a signature pack update which was provided to us in just 24 hours.

All our “false negative” (modified exploit) cases were detected correctly, and none of the false positive test cases triggered. Resistance to known evasion techniques was also excellent, with Proventia being one of the few products to collect a clean sheet across the board in our evasion tests. Not only were the fragmented and obfuscated attacks blocked successfully, but every one of them was decoded accurately as well. This is the level of performance to which we would like to see all IDS and IPS products aspire.

Usability

With the removal of the aged *Workgroup Manager* Console and its replacement with the more scalable and flexible *SiteProtector*, Proventia has taken a huge leap forward.

The only significant portion of “legacy” code remaining in the user interface is the Policy Editor, though this is due to be replaced in the not too distant future with the new Java-based *Common Policy Editor.* This, as its name implies, will provide a common policy management interface across the entire Proventia range.

The old Workgroup Manager Console was one of the most comprehensive and easy to use out of the box and for a long time was the benchmark against which other IDS consoles were measured. With SiteProtector, ISS has done it again, producing one of the best consoles we have seen in our labs to date. It is worth pointing out that this is no longer an extra cost option, but is included in the price of the sensor (the SecurityFusion and new Management Reporting modules remains extra-cost options, however).

Sensor and policy management are amongst the most straightforward and scalable that we have seen to date. Policies can be applied to multiple sensors at the click of a mouse, and the use of rules to auto-group assets together with the ability to apply policies at site or group levels (manually or via automated "subscription") means it is very easy to install and activate sensors with little or no administrator intervention. Currently, Proventia is probably one of the easiest products to deploy across a large, distributed network that we have seen.

Alert handling, too, is excellent, with SiteProtector attempting to simplify the life of the administrator via automatic impact analysis and event correlation across vulnerability assessment tools and network sensors.

The result should be the ability to reduce the number of critical alerts appearing at the Console to a more manageable level in even the largest installations by grouping them together and tracking them as "incidents". The ability to report and annotate at an incident level is useful, as is the ability to **manually** correlate multiple events into a single incident if required.

Infinitely customisable views provide the means to drill down into the event data from almost any angle via the *Sensor Analysis* tab in the Console. The resulting views can be saved and recalled on demand or run at regular intervals via a scheduler, and the Dashboard provides high level graphical summaries. It is nice to see a company producing a genuine **advance** in reporting and analysis tools rather than relying purely on Crystal Reports and a few basic pre-defined templates.

Having said that, the latter is now used to flesh out the reporting capabilities by providing a number of pre-defined management reports as an extra-cost option, thus covering all possible reporting angles.

Contact Details

Company name: Internet Security Systems, Inc.

E-mail: sales@iss.net

Internet: www.iss.net

Address:
6303 Barfield Road, 4th Floor
Atlanta, GA 30328
USA

Tel: +1 404-236-2600 or 1-800-776-2362
Fax: +1 404-236-2614

APPENDIX A – TEST RESULTS

The aim of this procedure (based on V3.0 of the NSS Group IDS Testing Methodology) is to provide a thorough test of all the main components of a Gigabit IDS device in a controlled and repeatable manner, and in the most "real world" environment which it is possible to simulate in a test lab.

The Test Environment

The network is 100/1000Mbit Ethernet with CAT 5e cabling and Cisco Catalyst 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. There is no firewall protecting the target network.

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.

All "normal" network traffic, background load traffic and exploit traffic crossing the switches is mirrored to **two** SPAN ports on the Catalyst 6503 (the same traffic is mirrored to both simultaneously). The sensor's detection interface is connected to one SPAN port, whilst an Adtech network monitoring device monitors the same mirrored traffic via the second SPAN port to ensure that the total amount of traffic never exceeds 1Gbps (which would invalidate the test run).

The sensor is bound to the Gigabit network interface in "stealth mode" wherever that is supported (i.e. no IP address) and a separate interface is used to connect the sensor 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 logging 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 product, this test attempts to demonstrate how accurately the sensor detects and logs a wide range of common exploits, port scans, and Denial of Service attempts. All exploits are run with no load on the network and no IP fragmentation.

Our attack suite contains over 100 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.

The "**default**" *Attack Recognition Rating* (ARR) is expressed as a percentage of detected 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**" ARR is determined. This demonstrates how effective the vendor is at responding to a requirement for new or updated signatures.

Both the *default* and *custom* ARR 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.

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

The device attains a “PASS” for each test case if it does **not** raise an alert. 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.

- ***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 logging 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 a number of common basic attacks (our baseline suite) in their normal state, with no evasion techniques applied.

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

Test 2.2 - Packet Fragmentation and Stream Segmentation

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

- ***Test 2.2.1 -*** *IP fragmentation - ordered 8 byte fragments*
- ***Test 2.2.2 -*** *IP fragmentation - ordered 24 byte fragments*
- ***Test 2.2.3 -*** *IP fragmentation - out of order 8 byte fragments*
- ***Test 2.2.4 -*** *IP fragmentation - ordered 8 byte fragments, duplicate last packet*
- ***Test 2.2.5 -*** *IP fragmentation - out of order 8 byte fragments, duplicate last packet*
- ***Test 2.2.6 -*** *IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse*
- ***Test 2.2.7 -*** *IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new)*
- ***Test 2.2.8 -*** *IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old)*
- ***Test 2.2.9 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums*
- ***Test 2.2.10 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags*
- ***Test 2.2.11 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream*
- ***Test 2.2.12 -*** *TCP segmentation - ordered 1 byte segments, duplicate last packet*

- ***Test 2.2.13 -*** *TCP segmentation - ordered 2 byte segments, segment overlap (favour new)*
- ***Test 2.2.14 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers*
- ***Test 2.2.15 -*** *TCP segmentation - out of order 1 byte segments*
- ***Test 2.2.16 -*** *TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits*
- ***Test 2.2.17 -*** *TCP segmentation - ordered 1 byte segments, segment overlap (favour new)*
- ***Test 2.2.18 -*** *TCP segmentation - out of order 1 byte segments, PAWS elimination (interleaved dup segs with older TCP timestamp options)*
- ***Test 2.2.19 -*** *IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery*
- ***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 detected and an alert raised in **any** form, and (ii) 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 detected and an alert raised in **any** form, and (ii) 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 detected and an alert raised in **any** form, and (ii) 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 across the network 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).

- ***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 log new exploits when the 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" network) then opens various numbers of TCP sessions from 10,000 to 1,000,000 (one million) with the Spirent Reflector (on the "internal" network) and the sensor will be expected to track each of these legitimate sessions. 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 device 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 freshly-launched exploits once all the connections are open.

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 logging 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 - 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 3.3 - 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 - State Preservation:*** *As Test 3.2.3 following tuning*

Section 4 – Detection Performance Under Load

The aim of this section is to verify that the sensor is capable of detecting and logging 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 log** suspicious traffic - no session-termination techniques are employed (i.e. RST packets from the sensor).

Our "attacker" host launches a fixed number of exploits at a target host on the subnet being monitored by the sensor. The Adtech network monitor is configured to monitor the *same* traffic on a second 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 across the network 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 Detection Rate* (ADR) at each background load is expressed as a percentage of the number of exploits detected by the sensor against the number verified by the Adtech network monitor and target host.

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 across the network protected by the sensor. 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 throughout each test. Multiple tests are run and averages taken where necessary.

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

- ***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 logging 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% detection 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 – Stability & Reliability

These tests attempt to verify the stability of the device under test under various extreme conditions.

- ***Test 5.1.1 - 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 to the external network, and the ISIC target is located on the internal network protected by the sensor. ISIC traffic is transmitted across the network 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 logging exploits throughout the test to attain a PASS.*

Section 6 – 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 6.1 - 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 6.1.1 - Open ports:*** *We will scan the open ports and active services on the management interface and report on known vulnerabilities.*
- ***Test 6.1.2 - 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 sensor, and that interface is also the target. ISIC traffic is transmitted to the management interface of 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 (a) operational and capable of detecting and logging exploits, and (b) capable of communicating in both directions with the management server/console throughout the test to attain a PASS.*
- ***Test 6.1.3 -*** *We note whether the ISIC attacks themselves are detected by the sensor even though targeted at the management port.*

ISS Proventia A604 Test Results

Section 1 - Detection Engine

Test 1.1 – Attack Recognition	Attacks	Default ARR	Custom ARR
Test 1.1.1 - Backdoors	7	7	7
Test 1.1.2 - WINS/DNS	3	3	3
Test 1.1.3 - DOS	10	10	10
Test 1.1.4 - False negatives (modified exploits)	14	14	14
Test 1.1.5 - Finger	4	4	4
Test 1.1.6 - FTP	5	5	5
Test 1.1.7 - HTTP	43	42	43
Test 1.1.8 - ICMP	2	2	2
Test 1.1.9 - Reconnaissance	8	8	8
Test 1.1.10 - RPC	9	9	9
Test 1.1.11 - SSH	1	1	1
Test 1.1.12 - Telnet	1	1	1
Test 1.1.13 - Database	1	1	1
Test 1.1.14 - Mail	1	1	1
Test 1.1.15 - Voice	1	1	1
Total	**110**	**109 / 110**	**110 / 110**
		99%	**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	PASS	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	**17 / 17**	**17 / 17**

Section 2 - IPS Evasion

Test 2.1 – Evasion Baselines	Detected?
Test 2.1.1 - NSS Back Orifice ping	YES
Test 2.1.2 - Back Orifice connection	YES
Test 2.1.3 - FTP CWD root	YES
Test 2.1.4 - ISAPI printer overflow	YES
Test 2.1.5 - Showmount export lists	YES
Test 2.1.6 - Test CGI probe (/cgi-bin/test-cgi)	YES
Test 2.1.7 - PHF remote command execution	YES
Total	**7 / 7**

Test 2.2 – Packet Fragmentation/Stream Segmentation	Detected?	Decoded?
Test 2.2.1 - IP fragmentation - ordered 8 byte fragments	YES	YES
Test 2.2.2 - IP fragmentation - ordered 24 byte fragments	YES	YES
Test 2.2.3 - IP fragmentation - out of order 8 byte fragments	YES	YES
Test 2.2.4 - IP fragmentation - ordered 8 byte fragments, duplicate last packet	YES	YES
Test 2.2.5 - IP fragmentation - out of order 8 byte fragments, duplicate last packet	YES	YES

Test 2.2.6 - IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse	YES	YES
Test 2.2.7 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new)	YES	YES
Test 2.2.8 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old)	YES	YES
Test 2.2.9 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums	YES	YES
Test 2.2.10 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags	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
Test 2.2.12 - TCP segmentation - ordered 1 byte segments, duplicate last packet	YES	YES
Test 2.2.13 - TCP segmentation - ordered 2 byte segments, segment overlap (favour new)	YES	YES
Test 2.2.14 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers	YES	YES
Test 2.2.15 - TCP segmentation - out of order 1 byte segments	YES	YES
Test 2.2.16 - TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits	YES	YES
Test 2.2.17 - TCP segmentation - ordered 1 byte segments, segment overlap (favour new)	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
Test 2.2.19 - IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery	YES	YES
Test 2.2.20 - TCP segmentation - ordered 16 byte segments, segment overlap (favour new (Unix))	YES	YES
Total	**20 / 20**	**20 / 20**

Test 2.3 – URL Obfuscation	**Detected?**	**Decoded?**
Test 2.3.1 - URL encoding	YES	YES
Test 2.3.2 - /./ directory insertion	YES	YES
Test 2.3.3 - Premature URL ending	YES	YES
Test 2.3.4 - Long URL	YES	YES
Test 2.3.5 - Fake parameter	YES	YES
Test 2.3.6 - TAB separation	YES	YES
Test 2.3.7 - Case sensitivity	YES	YES
Test 2.3.8 - Windows \ delimiter	YES	YES
Test 2.3.9 - Session splicing	YES	YES
Total	**9 / 9**	**9 / 9**

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

Section 3 - Stateful Operation

Test 3.1 – Stateless Attack Replay	**Alert?**	**Pass/Fail**
Test 3.1.1 - Stateless Web exploits	NO	PASS
Test 3.1.2 - Stateless FTP exploits	NO	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 - State Preservation	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	PASS	PASS	PASS	PASS	PASS	PASS
Test 3.3.2 - State Preservation	PASS	PASS	PASS	PASS	PASS	PASS	PASS

Section 4 - Detection/Blocking Performance Under Load

Test 4.1 – UDP traffic to random valid ports	150Mbps	300Mbps	450Mbps	600Mbps	Max
Test 4.1.1 - 256 byte packet test - max 270,000pps	100%	100%	100%	100%	600Mbps
Test 4.1.2 - 550 byte packet test - max 132,000pps	100%	100%	100%	100%	600Mbps
Test 4.1.3 - 1000 byte packet test - max 73,000pps	100%	100%	100%	100%	600Mbps

Test 4.2 – HTTP "maximum stress" traffic with no transaction delays	150Mbps	300Mbps	450Mbps	600Mbps	Max
Test 4.2.1 - Max 1500 connections per second - ave packet size 1000 bytes - max 73,000 packets per second	100%	100%	100%	100%	600Mbps
Test 4.2.2 - Max 3000 connections per second - ave packet size 540 bytes - max 135,000 packets per second	100%	100%	100%	100%	600Mbps
Test 4.2.3 - Max 6000 connections per second - ave packet size 440 bytes - max 165,000 packets per second	100%	100%	100%	100%	600Mbps
Test 4.2.4 - Max 12000 connections per second - ave packet size 360 bytes - max 198,000 packets per second	100%	100%	100%	100%	600Mbps

Test 4.3 – HTTP "maximum stress" traffic with transaction delays	150Mbps	300Mbps	450Mbps	600Mbps	Max
Test 4.3.1 - Max 3000 connections per second - ave packet size 540 bytes - max 135,000 packets per second - 10 sec delay - max 50,000 open connections	100%	100%	100%	100%	600Mbps
Test 4.3.2 - Max 6000 connections per second - ave packet size 440 bytes - max 165,000 packets per second - 10 sec delay - max 50,000 open connections	100%	100%	100%	100%	600Mbps

Test 4.4 – Protocol mix	150Mbps	300Mbps	450Mbps	600Mbps	Max
Test 4.4.1 - 72% HTTP (540 byte packets) + 20% FTP + 6% UDP (256 byte packets). Max 2400 connections per second - ave packet size 540 bytes - max 129,000 packets per second - max 450 open connections	100%	100%	100%	100%	600Mbps

Test 4.5 – Real World traffic	150Mbps	300Mbps	450Mbps	600Mbps	Max
Test 4.5.1 - Pure HTTP (simulated browsing session on NSS Web site). Max 2800 connections per second - 12 new users per second - ave packet size 560 bytes - max 126,000 packets per second	100%	100%	100%	100%	600Mbps
Test 4.5.2 - Protocol mix - 72% HTTP (simulated browsing sessions as 2.5.1) + 20% FTP + 6% UDP (256 byte packets). Max 2200 connections per second - ave packet size 560 bytes - max 123,000 packets per second - max 900 open connections	100%	100%	100%	100%	600Mbps

Section 5 - Stability & Reliability

Test ID	Result
Test 5.1.1 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC	PASS

Section 6 - Management Interface

Test ID	Result
Test 6.1.1 - Open Ports	PASS
Test 6.1.2 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC	PASS
Test 6.1.3 - ISIC attacks detected against management interface?	NO

Section 1: Detection Engine

We installed one sensor with the latest signature pack, reporting to a single SiteProtector server. We used a modified version of the default *Attack Detector* policy, which had **all** attack signatures enabled (apart from port probes) as well as some key audit-only signatures.

Signature recognition (with blocking disabled) was excellent out of the box (99 per cent), and was increased to 100 per cent after the application of a signature pack update which was provided to us in just 24 hours.

We noted a minimum of “noise” in this release, with very few test cases raising multiple alerts for a single exploit. All our “false negative” (modified exploit) cases were detected correctly, demonstrating that the Proventia signatures are designed to detect the underlying vulnerability rather than a specific exploit wherever possible.

A major concern in deploying an IDS is the raising of false alarms. We noted no false positive alerts from our test suite. All in all, the Proventia continues to be one of the most consistently accurate IDS systems we have tested.

Section 2: IPS Evasion

Resistance to known evasion techniques was excellent, with Proventia being one of the few products to collect a clean sheet across the board in our evasion tests. *Fragroute, Whisker, ADMmutate, running exploits on non-standard ports* and even *RPC record fragging* all failed to trick Proventia into ignoring valid attacks.

Note that not only were the fragmented and obfuscated attacks detected successfully, but every one of them was decoded accurately as well. This is the level of performance to which we would like to see all IDS and IPS products aspire.

Section 3: Stateful Operation

Out of the box, the Proventia A604 handled over 1 million open connections easily. The sensor also continued to track state on our “half open” exploits beyond its configured maximum open connections, since Proventia is designed to ignore new connections once the limit is exceeded, but does not age out old ones. If you prefer old connections to be aged out in favour of allowing new ones when the state tables are full, ISS has added a parameter in the latest release which makes this behaviour configurable.

The Proventia appeared to be immune to stateless attack reply tools such as *Stick* and *Snot*.

Section 4: Detection/Blocking Performance Under Load

The Proventia A604 was tested up to 600Mbps, the rated speed of the appliance.

Performance under all load conditions was impeccable, and we would have no hesitation in rating the Proventia A604 as a true 600Mbps device as claimed by ISS.

Section 7: Stability & Reliability

The Proventia A604 performed consistently and mostly reliably throughout our tests.

Exposing the sensor interface to an extended run of ISIC-generated traffic had no adverse effect, and the device continued to detect and log all other exploits throughout and following the ISIC attack. There were no residual stability problems.

Section 6: Management Interface

Open ports on the management interface are restricted to 22/TCP (SSH), 901/TCP (SAMBA-SWAT) and 2998/TCP (ISS-REALSEC) to allow communication between the SiteProtector server and the Proventia appliance. Firewall rules can be put in place to ensure that connections are made from known management server(s) only. Once in place, port scans failed completely from any other PC on the management network.

The extended ISIC attack against the management interface had no effect on the appliance and its ability to detect and log attacks, and there was only a slight delay in communicating with the management server. However, no alerts were raised at any time to inform the administrator that the management port was under attack.

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