Radware DefensePro-3000 V2.43

Technical Evaluation

An NSS Group Report

First published May 2005 (Version 1.0)

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

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

TABLE OF CONTENTS

INTRODUCTION 1

Intrusion Prevention Systems (IPS) 1
- Host IPS (HIPS) 2
- Network IPS (NIPS) 2
- Rate-Based IPS (Attack Mitigator) 3

Detection Methods 3
- Pattern Matching 4
- Stateful Pattern Matching 4
- Protocol Decode 5
- Heuristic Analysis 7
- Anomaly Analysis 7
- Which Detection Method Is The Best? 7

Implementation Challenges 8
Requirements for effective prevention 9
The NSS Intrusion Prevention Group Test 10
- Performance 11
- Security Effectiveness 14
- Usability 16

RADWARE DEFENSEPRO-3000 V2.43 17

Executive Summary 17
Architecture 17
- DefensePro 17
- Web-Based Management Interface 20
- Command Line Interface (CLI) 20
- Configware Insite 20

Performance 21
Security Effectiveness 22
Usability 23
- Installation 23
- Configuration 25
- Policy Management 26
- Alert Handling 30
- Reporting and Analysis 32

Verdict 35
Contact Details 37

APPENDIX A – TEST RESULTS 38

The Test Environment 38
Section 1 – Detection Engine 38
Section 2 – Evasion 40
Section 3 – Stateful Operation 42
Section 4 – Detection/Blocking Performance Under Load 44
Section 5 – Latency & User Response Times 48
Section 6 – Stability & Reliability 50
Section 7 – Management and Configuration 50
Radware DefensePro-3000 V2.40 Test Results 52
- Section 1 - Detection Engine 52
- Section 2 - IPS Evasion 52
- Section 3 - Stateful Operation 54
- Section 4 - Detection/Blocking Performance Under Load 54
- Section 5 - Latency & User Response Times 55
- Section 6 - Stability & Reliability 55
- Section 7 - Management Interface 55

TABLE OF FIGURES

Figure 1 - DefensePro: Hardware architecture 18
Figure 2 - DefensePro: Web-Based Management 19
Figure 3 - DefensePro: Configware Insite 20
Figure 4 - DefensePro: First Time Wizard 24
Figure 5 - DefensePro: Configuring management access 25
Figure 6 - DefensePro: Creating policies in the Connect & Protect table 27
Figure 7 - DefensePro: Configuring SYN Flood Protection policy 28
Figure 8 - DefensePro: Configuring global security parameters 29
Figure 9 - DefensePro: Attack Log 31
Figure 10 - DefensePro: Reports 33
Figure 11 - DefensePro: Dashboard 34

The NSS Group

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

With British headquarters, and security and network infrastructure testing facilities in the South of France, The NSS Group offers a range of specialist IT, networking and security-related services to vendors and end-user organisations world-wide.

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

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

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

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

Foreword

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

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

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

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

It is worth pointing out that not every product submitted for testing receives an *NSS Approved* award. Standards are very high, and only those appearing in this report have received ***NSS Approved*** awards. For this latest edition, **ten** vendors submitted a total of **twelve** products for testing, and **eight** of these passed our stringent testing to receive **NSS Approved**. It is heartening to note that this is a much-improved success ratio over Edition 2.

We believe that our IPS test methodologies - which have been updated again for this test - will become the *de facto* standard for testing in-line Intrusion Prevention/Attack Mitigation devices, and the *NSS Approved* logo an essential item on the list of requirements when purchasing these products.

We also believe that this report is essential reading for anyone considering deploying Intrusion Prevention Systems in their networks, either in a test or live situation, and we hope that you find it both informative and useful in making your purchasing decisions. The latest **IPS Group Test** report can be viewed on-line at *www.nss.co.uk/ips*

Bob Walder

INTRODUCTION

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

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

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

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

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

Intrusion Prevention Systems (IPS)

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

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

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

Within the IPS market place, there are two main categories of product: *Host IPS* and *Network IPS*, with the latter being further sub-divided into *Content-Based* and *Rate-Based* (or *Attack Mitigation*) systems.

Host IPS (HIPS)

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

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

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

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

Network IPS (NIPS)

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

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

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

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

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

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

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

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

Rate-Based IPS (Attack Mitigator)

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

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

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

Detection Methods

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

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

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

Pattern Matching

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

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

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

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

Stateful Pattern Matching

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

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

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

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

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

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

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

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

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

Protocol Decode

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Heuristic Analysis

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

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

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

Anomaly Analysis

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

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

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

Which Detection Method Is The Best?

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

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

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

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

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

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

Implementation Challenges

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

If a passive IDS fails, the worst that can happen is that some attempted attacks may go undetected. If an in-line device fails, however, it can seriously impact the performance of the network.

Perhaps latency rises to unacceptable values, or perhaps the device fails closed, in which case you have a self-inflicted Denial of Service condition on your hands. On the bright side, there will be no attacks getting through! But that is of little consolation if none of your customers can reach your e-commerce site.

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

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

Dropped packets are also an issue, since if even one of those dropped packets is one of those used in the exploit data stream it is possible that the entire exploit could be missed.

Most high-end IPS vendors will get around this problem by using custom hardware, populated with advanced FPGAs and ASICs - indeed, it is necessary to design the product to operate as much as a switch as an intrusion detection and prevention device.

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

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

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

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

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

Requirements for effective prevention

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

- **In-line operation -** only by operating in-line can an IPS device perform true protection, discarding all suspect packets immediately and blocking the remainder of that flow

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

The NSS Intrusion Prevention Group Test

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

This exhaustive review will give readers a complete perspective of the capabilities, maturity and suitability of the products tested for their particular needs.

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

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

To assess the complex matrix of IPS/Attack Mitigator performance and security requirements, the NSS Group has developed a specialised lab environment that is able to exercise every facet of an IPS product. The test suite contains over 800 individual tests that evaluate IPS products in three main areas: *performance and reliability, security accuracy,* and *usability*.

This thorough review should give readers a complete perspective of the capabilities, maturity and suitability of the products tested for their particular needs.

Performance

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

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

Detection/Blocking Performance Under Load

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

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

All tests are repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic (or up to the maximum rated throughput of the device in 25 per cent increments should this be less than 1Gbps). The test is conducted with UDP, HTTP, and mixed-protocol traffic and includes packet rates up to 453,000 packets per second and connection rates up to 20,000 connections per second.

Latency & User Response Times

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

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

throughput = window / RTT

where *window* is the maximum TCP window size (64 Kbytes by default) and RTT is the round trip time in the network.

This equation tells us that the throughput of a TCP connection is inversely proportional to network latency (note that this is TCP throughput for *one* connection - the aggregate bandwidth is not affected by latency). In other words, if you double latency, you halve throughput.

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

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

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

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

Such critical value calculations are important when TCP connections achieve maximum throughput, which is true for large data transfers.

For smaller data transfers, and non-TCP applications like NFS, latency has a more direct impact on user experience - response time is directly proportional to latency. That is, *doubling latency doubles response time.* In these situations, the latency of the network in which a NIPS is deployed determines the acceptable latency of the NIPS.

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

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

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

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

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

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

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

This "*application latency*" is measured both with no background load and while the device is under attack.

Stability & Reliability

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

In the first part of this test, we expose the external interface of the sensor to a constant stream of attacks over an extended period of time. The device is configured to block and alert, and thus this test provides an indication of the effectiveness of both the blocking and alert handling mechanisms. A continuous stream of exploits mixed with some legitimate sessions is transmitted through the sensor at a maximum rate of 90 per cent of the claimed throughput of the device for eight hours with no additional background traffic.

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

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

We scan the management interface for open ports and active services and report on known vulnerabilities. We also stress the protocol stack of the management interface of the NIPS by exposing it to malformed traffic from the ISIC test tool. The device is expected to remain (a) operational and capable of detecting and blocking exploits, and (b) capable of communicating in both directions with the management server/console throughout the test to attain a PASS. We also note whether the sensor detects the ISIC attacks even though targeted at the management port.

Security Effectiveness

Detection Accuracy & Breadth

This group of tests verifies that the NIPS will not block legitimate traffic (*Accuracy*) and is capable of detecting and blocking a wide range of common exploits (*Breadth*). Although *breadth* is extremely important, *accuracy* is critical because a NIPS that blocks legitimate traffic will not remain in-line for long.

We have a number of trace files of normal traffic with "suspicious" content, together with several "neutered" exploits that have been rendered completely ineffective. The IPS attains a "PASS" for each test case if it does **not** raise an alert and does **not** block the traffic. Whilst it is not possible to validate completely the entire signature set of any IPS, this test demonstrates how accurately the IPS detects and blocks a wide range of common exploits, port scans, and Denial of Service attempts.

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

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

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

Resistance To Evasion Techniques

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

The tests consist of four parts (only the third is applicable to Rate-Based devices):

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

For each of the evasion techniques, we note if (i) the attempted attack is blocked successfully (the primary aim of any IPS device), (ii) the attempted attack is detected and an alert raised in **any** form, and (iii) if the exploit is successfully “decoded” to provide an accurate alert relating to the original exploit, rather than alerting purely on anomalous traffic detected as a result of the evasion technique itself.

Stateful Operation

If the IPS is tracking TCP session state, then it has the potential to introduce denial of service when the session table becomes full (too many connections) or if it can’t keep up with the creation of new sessions (too many connections per second).

As with latency and bandwidth, the number of connections supported by the IPS and its connection per second rate should be matched to the network.

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

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

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

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

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

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

Usability

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

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

RADWARE DEFENSEPRO-3000 V2.43

Executive Summary

Radware offers a range of DefensePro appliances from 200Mbps to 3Gbps throughput. The DefensePro-3000 is based on a layer 7 switching architecture, providing a high port density appliance, with protection for up to 11 segments in a single box, enabling multi-Gigabit protection across multiple network segments with a single device.

The DefensePro appliance performs bi-directional, stateful, deep packet inspection and hardware-accelerated signature matching to block intrusions. Dynamic traffic shaping ensures the continuity of mission critical applications by controlling end-to-end bandwidth to guarantee service levels. It also controls the bandwidth usage of various applications such as P2P, e-mail, Web, DNS, ERP, CRM and more.

Overall, the performance of DefensePro is very good. Although recognition rates and resistance to false positives could be improved, throughput and latency are excellent under almost all network loads and across all packet sizes. We also found DefensePro to be very stable, surviving our extended reliability tests without missing a beat, and without blocking any legitimate traffic or succumbing to common evasion techniques.

The device can be managed via a command-line interface (CLI), a basic Web-based management (WBM) utility, or via the more extensive *Configware Insite (CWIS)*. All configuration and log capabilities are supported by all management options. When using CWIS, management, alert handling and reporting are relatively restricted, with no central policy deployment and limited alert handling capabilities. A new version is under development with more extensive reporting capabilities.

Architecture

There are four main components to the Radware DefensePro system:

- *DefensePro Appliance*
- *Web-Based Management (WBM) interface*
- *Command Line Interface (CLI)*
- *Configware Insite central management system (CWIS)*

The DefensePro appliances can be managed via CLI, WBM or CWIS (or a mixture of the three), using standard and secure communication channels: Telnet or SSH, HTTP or HTTPS and SNMP V1, 2C and 3. All configuration and log capabilities are supported by all management options.

DefensePro

The DefensePro appliance is offered in three different models:

- ***DefensePro-3000 -*** *Designed for deployment at the core of large enterprise networks, data centres and carrier links. This device offers a maximum of 3Gbps throughput, with protection for up to three Gigabit Ethernet segments and eight Fast Ethernet segments*

- ***DefensePro-1000 -*** *Designed for deployment at the core of enterprise networks and data centres. This device offers a maximum of 1Gbps throughput, with protection for up to two Gigabit Ethernet segments and eight Fast Ethernet segments*
- ***DefensePro-200 -*** *Designed for deployment at the enterprise perimeter. This devices offers a maximum of 200Mbps throughput, with protection for up to two Gigabit Ethernet segments and eight Fast Ethernet segments.*

The device submitted for testing was the DefensePro-3000. Based on a layer 7 switch, this is a 1U appliance which sports 7 fibre Gigabit ports and 16 copper Fast Ethernet ports on the front panel. Management can be via any of the detection ports, or via the dedicated serial console port.

There is no redundancy built in to the device (other than dual power supplies), and nor is there any High Availability solution on offer at present. Radware does, however, offer optional copper and fibre bypass devices to allow traffic to pass unimpeded should the DefensePro appliance fail.

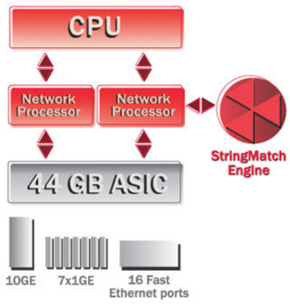

Figure 1 - DefensePro: Hardware architecture

Inside the appliance is 512MB of RAM, together with the four main components of the DefensePro hardware architecture:

- *Switching Fabric and Switching ASICs*
- *Network Processors*
- *StringMatch Engine*
- *Power PC RISC Processor*

44 Gbps Switching Fabric

DefensePro's non-blocking 44 Gigabit backplane is based on a multi-layered distributed switching architecture using ASICs that ensure wire speed switching for the seven Gigabit ports and 16 Fast Ethernet ports.

Network Processors

The two network processors work in parallel and are capable of processing multiple packets simultaneously to provide accelerated layer 4-7 security switching. Between them they handle all tasks related to packet processing - including traffic forwarding and blocking, traffic shaping, and delayed binding for protection against SYN flooding - all at multi-Gigabit speeds.

Prevention of Denial of Service attacks and SYN floods can be performed at a rate of up to 1 million SYNs per second. Detection and protection against layer 4 exploits is completed by the network processors and helps boost the performance when protecting against these type of exploits.

If deeper packet inspection is required (i.e. layer 7 scanning), then packets are forwarded to the StringMatch Engine - a dedicated hardware card designed specifically to provide accelerated signature and pattern matching - for signature identification. Based on the StringMatch pattern matching result, which determines whether the packet is legitimate or part of a malicious attack, the network processor either forwards the packet or drops it and resets the session.

In addition to cleaning all suspect traffic, the network processors enable end-to-end traffic shaping, managing bandwidth allocation to ensure continuous service levels for all secure traffic, guaranteeing the continuity and QoS of mission critical applications even under attack.

Radware StringMatch Engine

The Radware StringMatch Engine is a dedicated hardware card designed specifically to provide accelerated deep packet inspection and signature matching. The StringMatch Engine consists of up to 8 ASICs, enabling 256,000 parallel string searches, and a high end Power PC RISC processor for scheduling and running the parallel search algorithms. Theoretically, the StringMatch engine provides 9 Gigabits of free-range searches and 16 Gigabits of fixed offset searches for unmatched performance.

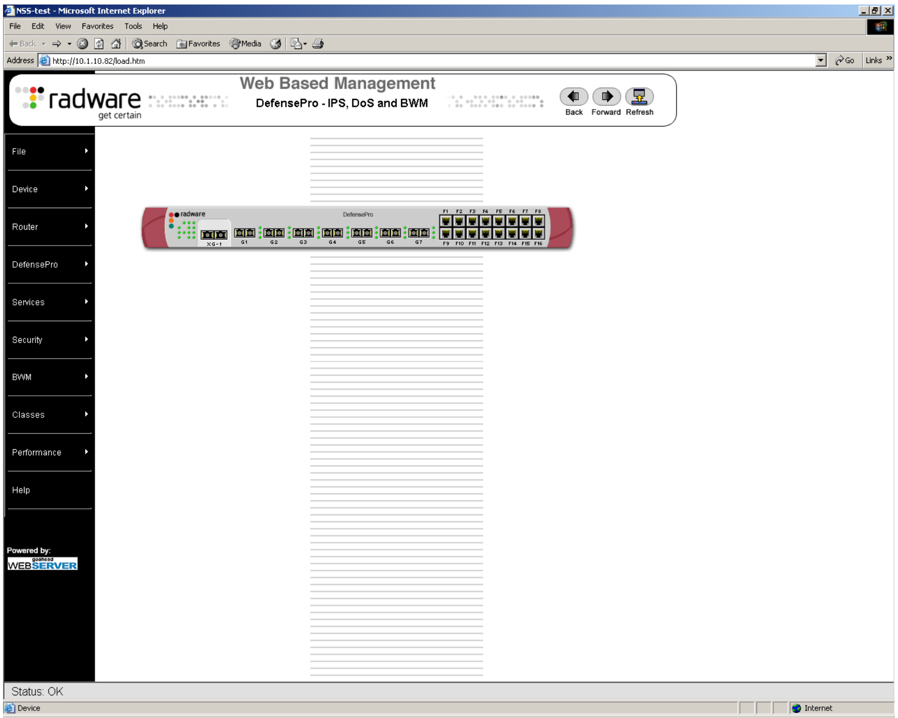

Figure 2 - DefensePro: Web-Based Management

CPU

DefensePro sports a 1GHz RISC processor Motorola PPC 7457 to manage and prioritise all security sessions.

The CPU identifies all current attacks and controls active operations across the StringMatch Engine and the Network processors to isolate, block and prevent attacks, while overseeing all security updates and networking requirements.

Web-Based Management Interface

Each DefensePro appliance can be managed directly via a Web-Based Management (WBM) interface running on the built-in Web server.

This provides access to all the main management, configuration, and alert handling functions via a browser-based GUI which can be used to connect to, and manage, a single device at a time.

Command Line Interface (CLI)

A text-based CLI is provided for direct management via the serial console or Telnet/SSH connections.

All configuration operations can be performed via the CLI, including attacks filter creation, policy configurations, etc. Reports are also available for CLI users, including top attack reports and detailed event logs.

Configware Insite

Configware Insite is a Java-based site-wide SNMP software management tool that is designed to enable unified administration, visibility and control of IP application performance and security across the enterprise.

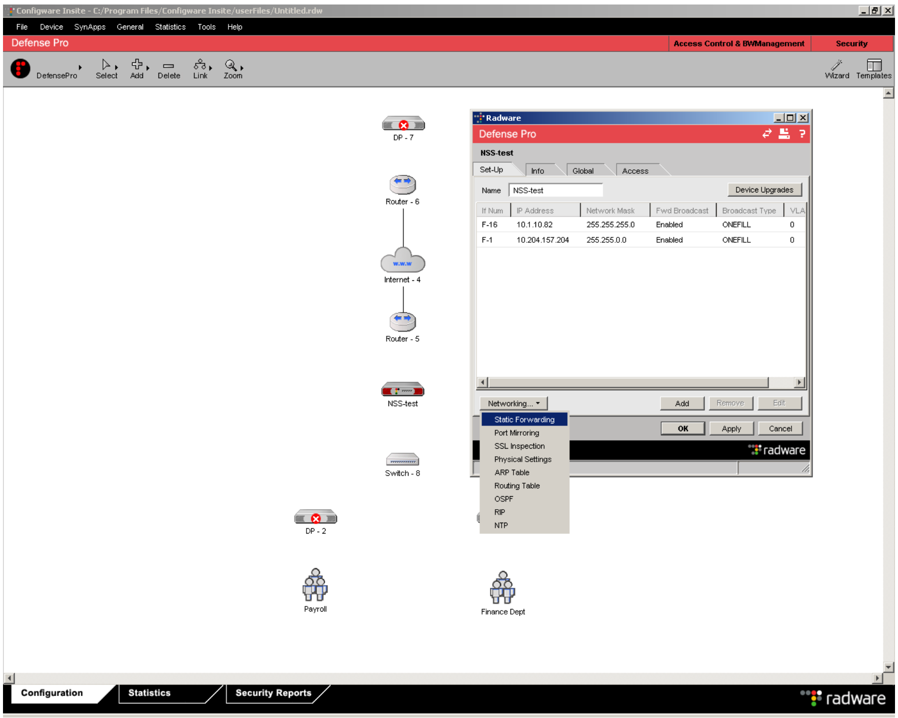

Figure 3 - DefensePro: Configware Insite

Based on an easy-to-use site map interface, Configware Insite allows the administrator to draw his network on-screen, configure Radware Intelligent Application Switching devices, and set-up SynApps ‘Application Aware’ Services, to address end-to-end IP application service requirements. Devices are simply placed, connected and configured in a visual environment to create site-wide topologies, establish device redundancies and set-up device networking parameters.

Configware Insite's statistics module provides real-time and historical views of actual application performance levels for monitoring site wide operations and simple pinpointing of vulnerabilities and failures. This affords complete visibility and control over the performance of Web and application servers, security tools, cache servers, and Internet links.

Configware Insite enables the configuration of SynApps application aware services, extending control over application requirements across layers 4-7 including:

- ***Health Monitoring & Traffic Redirection****: Establishing performance criteria and thresholds for device health and set-up of network traffic redirection and routing preferences.*
- ***Load balancing****: Configuration of enterprise traffic flow and device utilization settings.*
- ***Bandwidth Management****: Creation of bandwidth service level policies to prioritize traffic by applications and users.*
- ***DoS Protection and Intrusion Prevention****: Configuration of real-time security checks and attack signature registry.*

Within this context, DefensePro appliances can be added to the Configware Insite site map, following which they can be configured and managed along with all other Radware devices on the network. Security alerts are handled in the same way as other device alerts on the Configware Insite screen.

Configware Insite can operate in two-tier architecture or three-tier using *client-server mode*. The client-server mode allows system administrators to manage remote Radware devices while reducing the amount of SNMP traffic flowing through the WAN.

Performance

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

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

The DefensePro is rated by Radware at 3Gbps, and was tested to 1Gbps in this test. It turned in a good performance in almost all the tests, indicating that it can easily handle 1Gbps (and more) of normal network traffic.

DefensePro detected and, more importantly, blocked all attacks even when subjected to extreme loads, and under all other load conditions it performed well. At the more extreme loads (approaching 1Gbps at the higher connection rates), the device did exhibit slightly higher HTTP response times, and the occasional failed TCP connection. We also noted an inability to process the full 20,000 connections per second at 1Gbps, or the full 10,000 *delayed* connections per second at 1Gbps (see Test Results section for full details).

Despite this, we would rate DefensePro as a true 1Gbps device.

DefensePro's basic latency figures were excellent across the board under all traffic loads (probably due to the fact the device is based on a switching architecture), ranging from 117μs with 250Mbps of 256 byte packets, to 201μs with 1Gbps of 1000 byte packets.

Behaviour throughout the tests with no background traffic was consistent and predictable, with minimal increases as additional network load was applied from 250Mbps to 1Gbps. There were also minor increases when placing the device under a half load of 500Mbps of HTTP traffic, rising from 117μs to 160μs with 256 byte packets, 140μs to 183μs with 550 byte packets, and from 179μs to 219μs with 1000 byte packets.

100Mbps of SYN flood traffic was barely registered by the device, resulting in negligible (less than 10μs) increases in latency compared with the base figures. HTTP response times were very good overall and, once again, the addition of a 100Mbps SYN flood attack had a negligible effect on the performance.

Overall, latency figures were considered to be excellent for a device of this type under all load conditions and packet sizes. Clearly this device can be placed anywhere on the corporate network - from the perimeter to a heavily-loaded high-speed backbone - without significantly impacting overall network performance in any way.

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

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

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

Security Effectiveness

We installed one sensor with the latest updates, and enabled all signatures except for *Protocol Anomalies* and the *Archive* group (i.e. retired signatures).

Signature recognition was improved to 88 per cent following the application of a signature update after 24 hours, increased from a barely adequate 71 per cent out of the box. Blocking performance was one per cent higher throughout, due to one exploit being consistently blocked without an alert being raised. We consider this level of performance to be only just acceptable.

Performance in our "false negative" tests was poor out of the box, and although it improved following the signature update, there were still five misses out of the 14 test cases. This could indicate that many signatures are written for specific exploits rather than for the underlying vulnerability – perhaps an over-reliance on basic pattern matching rather than protocol decode.

A major concern in deploying an IPS is the blocking of legitimate traffic. All the tests passed successfully upon signature file update, although DefensePro turned in a less than perfect performance out of the box, failing in 5 out of 17 test cases.

Resistance to known evasion techniques was very good, with the DefensePro achieving a clean sweep across the board in most of our evasion tests.

Fragroute and *Whisker* both failed to deceive the device into ignoring valid attacks, and many of the attempts were decoded accurately. Of the miscellaneous evasion techniques, changing ports on Trojan programs and using RPC fragmentation both proved troublesome.

Out of the box, Radware claims that DefensePro can handle approximately 1,100,000 open connections with IP and TCP reassembly disabled (the default is 800,000). We did not attempt to verify this in our tests since we believe such anti-evasion features should always be enabled. We were able to verify up to 500,000 connections without tuning, but it was not possible to increase this to 1 million, since the device did not have enough memory to support this level of open connections with IP and TCP reassembly enabled. We also felt that the session ageing time was too low, causing state to be lost too early.

Stateless "exploits" are not alerted upon (this is correct behaviour in order to be resistant to *Stick* and *Snot* tools) and mid-flows are blocked by default (a mid-flow violation alert is raised). It is, however, possible to configure the device to allow mid-flows, and there is a configurable "grace period" where they are not enforced following a power-cycle to prevent blocking of legitimate traffic should the device come on-line in mid session.

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

Initial configuration of networking parameters is carried out via the serial console, following which the management software is installed.

In addition to the text-based command-line interface (CLI), DefensePro can be managed via the built-in *Web Based Management* (WBM) interface, which provides immediate and very comprehensive graphical two-tier management for a single device over a standard HTTP/HTTPS connection.

However, *Configware Insite* provides more extensive multi-device management and reporting capabilities via SNMP. Rather than focusing on a single device, Configware Insite presents the entire network configuration in a graphical format (the network diagram can be created on-screen), with settings and configuration options organised in a logically related manner.

Once the software has been installed, the DefensePro *First Time Wizard* enables the administrator to configure a blank DefensePro device from scratch with relevant protection polices that match his required network design and deployment scenario.

As the Wizard progresses, the administrator is prompted to enter the port configuration and the security policy to use initially, selected from a list including *corporate gateway, DMZ, DMZ mail, DMZ Web, corporate LAN, carrier/POP,* or *university LAN*. Each of these policies provides a basic configuration of signatures/filters which Radware feels is suitable for the intended deployment scenario.

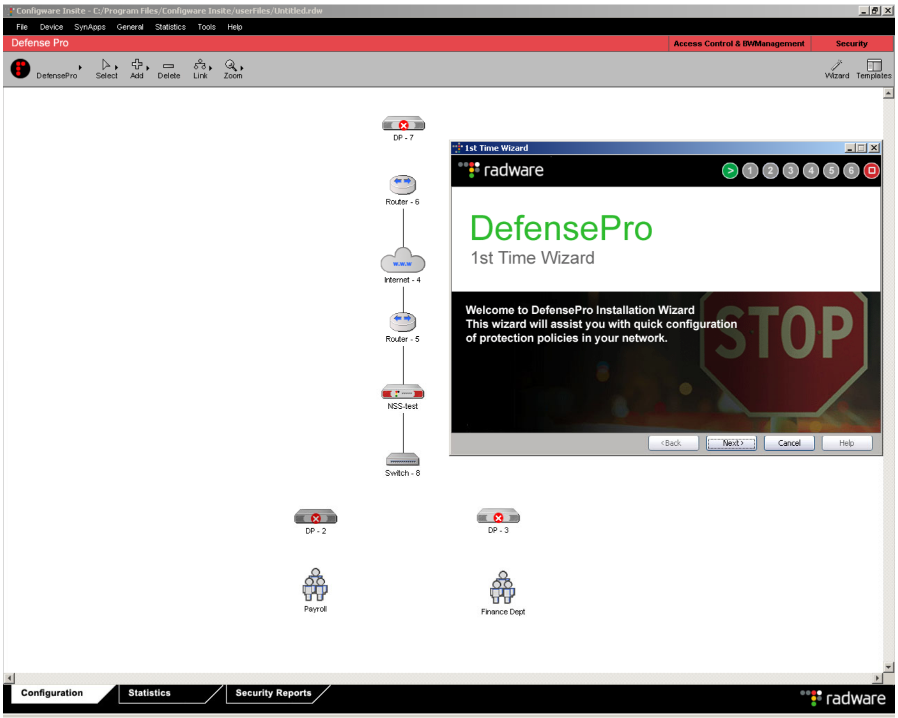

Figure 4 - DefensePro: First Time Wizard

Finally, the administrator specifies the network and VLAN configuration, and the reporting target (remote or local database where alerts are stored for analysis). The device is then ready to connect to the network and is enabled using a policy suitable for the deployment scenario specified in the Wizard.

In *Static Forwarding* mode DefensePro functions as a completely transparent network device. Scanning ports have a one-to-one forwarding ratio, where the traffic that comes from the receiving port is always sent out from its corresponding transmitting port. The ports are paired, meaning one port receives traffic while another transmits traffic. For each pair of ports the administrator selects the physical inbound port and the physical outbound port, plus the operation mode of the pair.

There are two available operation modes: *Switch* and *Process*. When a port pair is set to operate in the *Switch* mode, the traffic is switched straight through the device without any inspection. When the ports are set to operate in the *Process* mode, the traffic passes through the inspection engine where it is inspected for attacks, bandwidth control, and so on.

The processing of the traffic is performed by means of the various Bandwidth Management and Security filters. Both the inbound and outbound traffic to the organisation are processed, allowing the application of security policies and traffic shaping rules on traffic in both directions.

Scanning/detection ports are “invisible” to the network and thus can not have IP addresses . Any of the other physical ports on the device can be configured as a management port and will have an IP address allocated. Traffic received on the management port is not forwarded to any other ports, but is handled by the device itself.

A detailed *User Guide* is provided in electronic format only. This document is very comprehensive, and appears to be accurate and well written. The User Guide provides far more than basic instructions on using the GUI. It also offers plenty of background information covering the functions of the various options and parameters in depth, as well as good advice on deployment.

Configuration

Although DefensePro includes both a CLI and a direct basic Web-Based Management system as part of the product, this paper will concentrate purely on the more advanced *Configware Insite* product, which is also included out of the box.

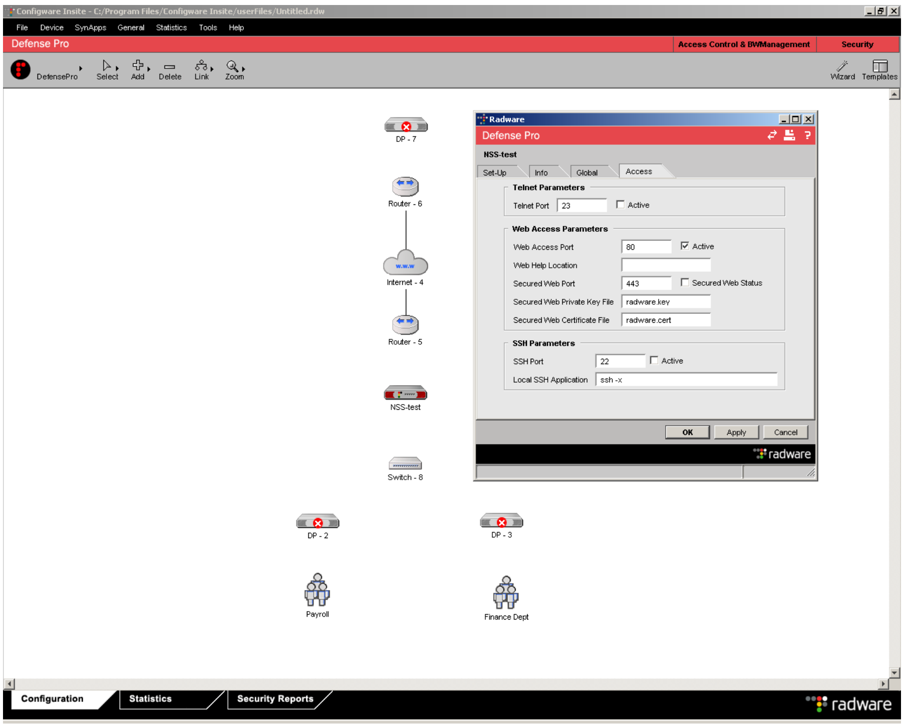

Figure 5 - DefensePro: Configuring management access

On first entering *Configware Insite* the administrator is presented with a graphical display of the site, which can be populated with icons of switches, routers, and other network elements as well as DefensePro units. These can be linked together to highlight physical or logical network links, and any of the DefensePro devices can be managed from here providing the administrator is authorised to do so.

However, it is still necessary to connect to individual devices in order to manage them or view alerts and reports – no consolidated view of all devices is available, and nor is it possible to define a single policy and push to all devices simultaneously or in groups. Site layouts can be saved for later recall.

Any number of users can be defined, and these can be allocated to Configware Insite as a whole (to allow use of the management interface) and to individual DefensePro appliances within the site. Thus separate administrators can be assigned different IPS devices to manage. Two levels of granularity only are available for user accounts: *administrator* and *operator*.

New DefensePro devices can be added quickly and easily in the site map by defining the IP address and port parameters (which ports are used for detection, management, etc.). The administrator defines which communications services are running on the management port (Telnet, HTTP, HTTPS or SSH), and can also configure basic device monitoring (checking availability of devices) with alerts appearing on the site map.

The *Signature File Update* feature provides constant updates of the Signature database (this is an extra-cost maintenance option). During the update process Configware Insite connects to the Radware Web site to acquire the file for the specified device. An updated Signature file can be found on the Radware website every Monday, though the site can be updated on any other day if an emergency update is required.

Updating of the Signatures file can be performed via an automatic download and update process, or can be performed manually on demand.

Policy Management

Protection policies are defined in the *Connect & Protect* table. This has a number of rows, giving it the appearance of a typical firewall rules table, and a set of global configuration parameters that apply across all policies.

The following general security settings can be applied in the *Security Parameters* window:

- ***Application Security -*** *A mechanism that provides advanced attack detection and prevention capabilities, checking the traffic on a packet-by-packet basis. This mechanism is used by the following security modules to provide maximum protection for network elements, hosts and applications: Intrusions, Anomalies, Application Security for DoS/DDoS. TCP reassembly can be enabled/disabled here.*
- ***DoS Shield -*** *The DoS Shield mechanism implements the sampling algorithm, and accommodates traffic flooding targeted to adversely affect network services. This mechanism is included in the DoS/DDoS security module.*
- ***Protocol Anomaly Protection -*** *The Protocol Anomaly Protection parameters are the general parameters of the Anomalies security module. Parameters include maximum URI length, minimum fragmented URI packet size, and minimum fragment size*
- ***Reporting –*** *Configures how and where alerts should be sent. It is possible to configure the reporting interval, maximum number of alerts to transmit per report (to prevent flooding of the console), and the transmission protocol (SNMP traps, e-mails, log entries, console echo).*

- ***Packet Reporting –*** *How and where raw packet data should be sent*
- ***IP Fragments –*** *Configures the IP fragmentation reassembly status and the minimum fragment protection status.*

Every row in the *Connect & Protect Table* represents a policy. A security policy contains security profiles that are activated within predefined ranges of ports/VLANs, or within a predefined network. The first task, therefore, is to define the scope of each policy in terms of IP address range, VLAN tag, inbound or outbound traffic, and so on.

Although this may seem trivial, it does actually give rise to a very powerful feature of the DefensePro system, since it is possible to define many different policies and have each one apply to only a subset of the protected network (right down to individual hosts, if required). For example, it would be possible to define a global policy which applies every protection feature, and then turn off DDOS protection and anti-scanning protection for a subset of hosts.

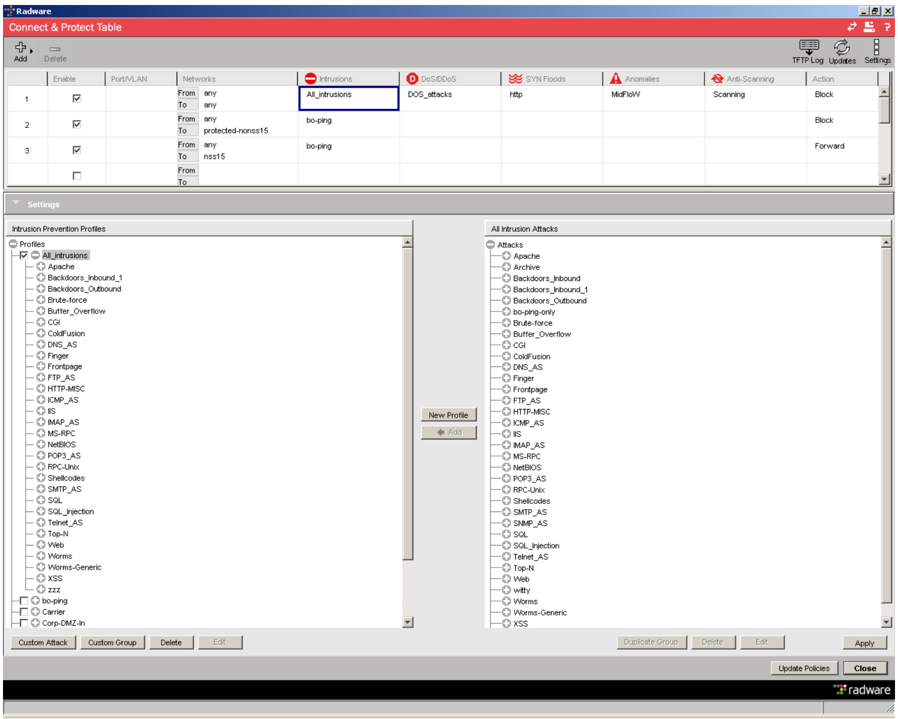

Figure 6 - DefensePro: Creating policies in the Connect & Protect table

Once the scope has been defined, the rest of the policy row is divided into columns representing the various security subsystems:

- *Intrusion module*
- *DoS/DDoS module*
- *SYN Flood module*
- *Anomaly module*
- *Anti-Scanning module*

Clicking within any of the security modules allows protection profiles to be added to that module to process traffic in a specific way. One or more profiles can be created for each security module and the profiles can then be associated with a policy.

The administrator is able to choose from an extensive range of signatures to add to a profile, and signatures are grouped together to make this process easier – for example, providing the ability to create a "*Web*" profile, and then select three or four groups of signatures dealing purely with Web exploits.

Where the pre-defined groups are not acceptable, the administrator can duplicate them and edit the duplicates to add or remove signatures as required. Unfortunately, there is no search facility, making it difficult to identify and locate individual signatures or groups of signatures when creating custom groups.

Once a group has been added to a profile, it is no longer possible to enable or disable individual signatures within it. Instead, it is possible only to enable or disable entire groups within the profile.

This is a shame, since it makes it very difficult to fine tune profiles without having to completely recreate the groups of signatures which comprise them. It would be preferable for each group and each signature within a group to have a check box against it in order to be able to enable or disable as required within a profile.

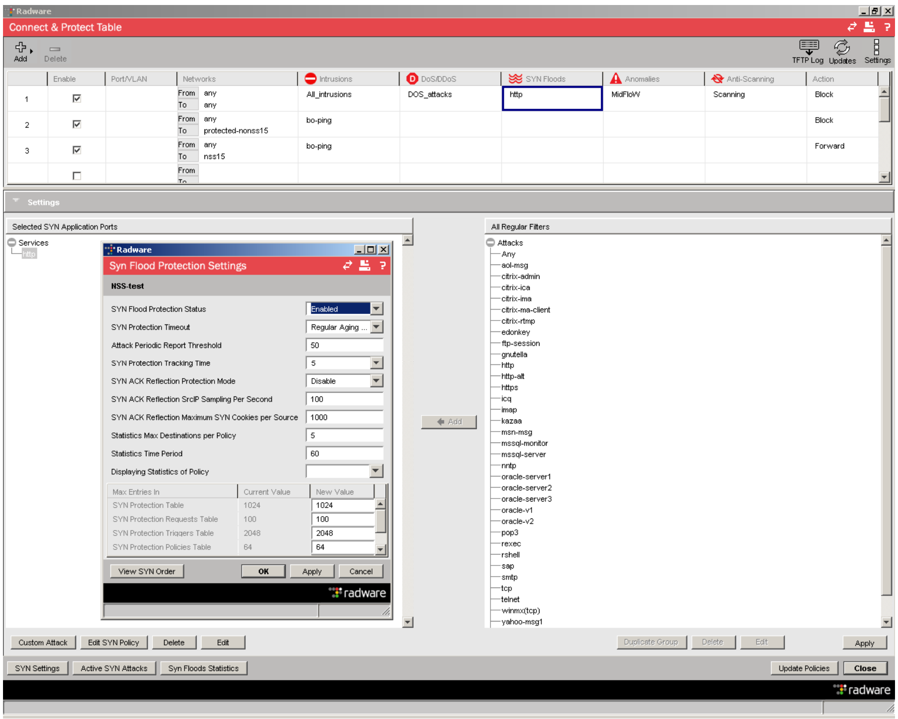

Figure 7 - DefensePro: Configuring SYN Flood Protection policy

Individual signatures have a range of editable parameters:

- ***Tracking Time -*** *Sets the amount of time (in milliseconds) in which a Threshold is measured (where applicable)*
- ***Threshold -*** *Sets the maximum number of attack packets that are allowed in each Tracking Time unit (where applicable)*
- ***Tracking Type -*** *Defines how the device decides which traffic to block or drop, when under an attack of this type. Values can be:*
    - ***Drop All:*** *Drop all packets, for when each packet of the defined attack is harmful.*

    - ***Target Attack:*** *For when the defined attack is destination-based, meaning the hacker is attacking a specific destination such as a Web server.*
    - ***Source and Target Attack:*** *For when the attack type is a source and destination based attack, meaning the hacker is attacking from a specific source IP to a specific destination IP.*
    - ***Source Attack:*** *For when the defined attack is source-based, meaning the attack can be recognized according to its source address.*

- ***Packet Report -*** *To specify capture of raw packet data*
- ***Action Mode -*** *When an attack is detected, one of the following actions can be taken:*
    - ***Report Only:*** *The packet is forwarded to the defined destination.*
    - ***Drop:*** *The packet is discarded.*
    - ***Reset Source:*** *Sends a TCP Reset packet to the packet Source IP.*
    - ***Reset Destination:*** *Sends a TCP Reset packet to the destination address.*
    - ***Reset Bi-directional:*** *Sends a TCP Reset packet to both the source and destination IP.*
- ***Risk -*** *Attack risk can be defined as High, Medium, or Low depending on the severity of the damage that the attack can cause*

These can be set on a per-signature basis, but unfortunately there is no way to apply bulk edits – say amending all IIS signatures to *High* severity. Conversely, the ability to drop a bad session (i.e. drop malicious packet and then mark the remaining session as bad) is only available as a global setting, and cannot be set on a per-signature basis. This should be made available as an additional *Action Mode* alongside the *Drop Packet* option.

Custom signatures can be created from scratch if required, and assigned to custom groups before being added to a profile. Signature definition is not for the faint-hearted, but Radware has produced a good interface here which makes it as straightforward as it is possible to be.

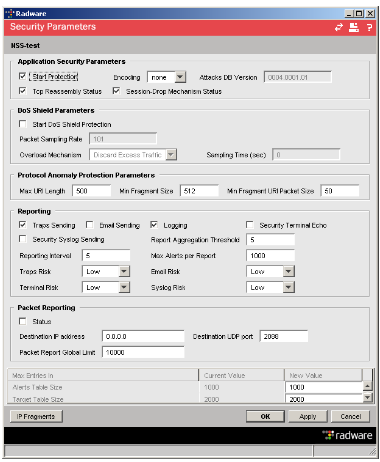

Figure 8 - DefensePro: Configuring global security parameters

Once Profiles have been created for each of the security modules required, the overall *Action* parameter can be specified for the Policy:

- ***Block -*** *The packet is identified as an attack and the action taken to prevent the attack is the action that was defined in the Block Action parameter of each security module.*
- ***Forward -*** *The packet is forwarded to the defined destination*
- ***Mixed –*** *When the Action parameter of a security module is changed using Web Based Management, the Action mode may appear as Mixed, since it is possible to merge Policies with different Action parameters into a single line in the Connect & Protect table. This seems to be unnecessarily confusing, and it would be preferable for each line of the Connect & Protect policy to remain separate.*

As mentioned before, multiple Policies can be defined within the same Connect & Protect table, each with different signature Profiles and actions, and each applying to different hosts or ranges of IP addresses. This is an extremely powerful and flexible system that provides very fine-grained control over a corporate security policy.

It is a shame that in larger deployments it is not possible to define a single policy and then apply to multiple devices. This would seem to be a basic requirement in an enterprise IPS management platform.

Alert Handling

When an attack is detected, the device creates a security event that includes the information relevant to this specific attack. Once an event has been created, the device reports it using several optional channels:

- ***Security Logs -*** *saved in flash memory (this log file can also be downloaded and examined within the management GUI if required)*
- ***SNMP traps -*** *can be sent to Configware Insite and a management station*
- ***Syslog messages -*** *can be sent to a Syslog server*
- ***E-mail messages -*** *can be sent to specific users*
- ***Security Terminal Echo -*** *echoed to appliance command line*

The **Attack Log** screen is divided into three panes. The alerts appear in the upper right pane, and selecting any alert displays detailed information about the exploit in the lower right pane. The pane running down the left hand side of the screen contains a number of pre-defined filters allowing the administrator to quickly and easily restrict the display to alerts classified as *High, Medium* or *Low Severity, Anomalies, Anti-Scanning, DOS* or *Intrusions*.

This pane also allows the definition of simple custom filters which can be stored and applied by checking the boxes next to them. A nice touch here is that multiple custom filters can be applied by checking more than one box, thus it is possible to build more complex filters in this way (although complex combinations of filters cannot be saved for re-use later).

Each alert entry shows the following information:

- ***Risk -*** *The attack severity level: high, medium or low.*
- ***Date/Time -*** *The date and time when the alert was generated.*
- ***Attack Name -*** *The name of the attack that was detected.*
- ***Physical Port -*** *The actual port on the device on which the attack arrived.*

- ***Action -*** *The reported action can be:*
  - ***Block:*** *The packet is identified as an attack and the action taken to prevent the attack is the action that was defined in the Block Action parameter of the security module.*
  - ***Forward:*** *The packet is forwarded to the defined destination.*
- ***Category -*** *The category of the attack: Anomalies, Anti-Scanning, DOS, or Intrusion.*
- ***Protocol -*** *The transmission protocol used to send the attack: TCP/ UDP/ ICMP/IP.*
- ***Source Address -*** *The IP address from which the attack arrived.*
- ***Source Port -*** *TCP/UDP source port.*
- ***Destination Address -*** *The IP address to which the attack is destined.*
- ***Destination Port -*** *TCP/UDP destination port.*
- ***Radware Attack ID -*** *Radware's unique identifier of the attack.*
- ***Packet Count -*** *The number of packets in the attack*
- ***Packet Bandwidth -*** *The bandwidth of the attack since the latest trap was sent (KByte).*

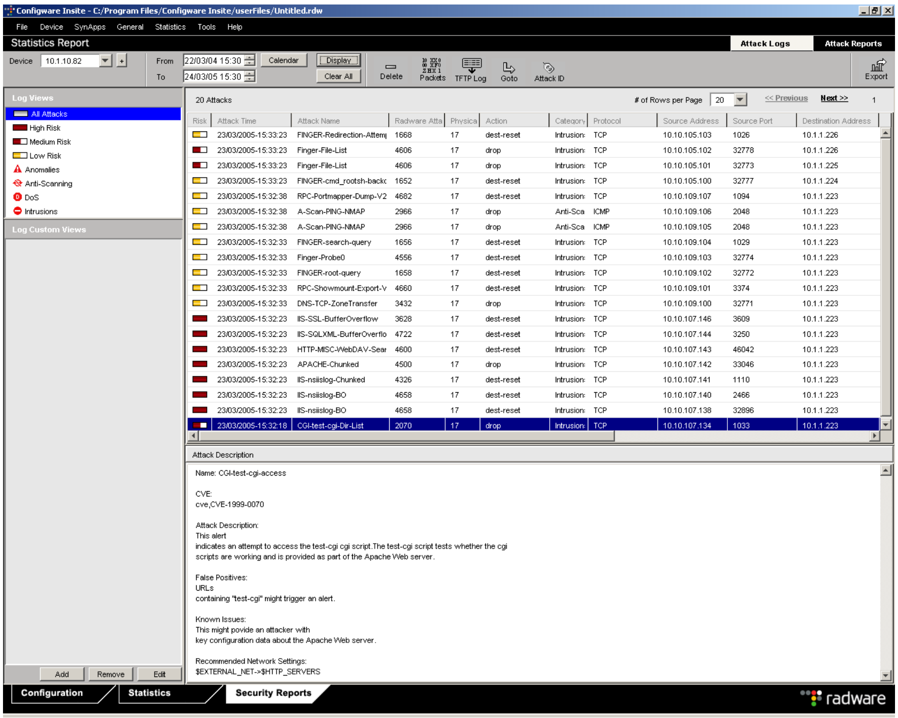

Figure 9 - DefensePro: Attack Log

- ***Status -*** *The current status of the event. For Intrusions, Anomalies, Anti-Scanning attacks, and Application Security for DoS/ DDoS attacks the following status can appear:*
  - ***Occurred:*** *Each packet matched with signatures is reported as an attack and must be dropped.*
  - ***Started/Terminated:*** *When the number of packets that match with signatures, goes beyond the predefined threshold within the Tracking Time, the reported Attack Status is "Started". When the number of packets that match with signatures is below the predefined threshold, the reported Attack Status becomes "Terminated".*

  - ***Ongoing:*** *A status report which occurs between "Started" and "Terminated".*

For DoS Shield attacks the following status indicators can appear:

  - ***Alert:*** *This status is reported when the number of packets that match with signatures goes beyond the predefined Warning Threshold.*
  - ***Active:*** *This status is reported when the number of packets that match with signature, goes beyond the predefined Activation Threshold.*
  - ***Block:*** *This status is reported when the number of packets that match with signatures goes beyond the predefined Drop Threshold.*
  - ***De-al:*** *The Deactivation Alert status is reported when the attack is about to be terminated.*
  - ***De-ac:*** *The Deactivation status is reported when the attack is terminated.*
- ***Device IP -*** *The IP of the device with which the attack is associated.*
- ***VLAN Tag -*** *VLAN Tag information*

When configured, it is also possible to view the raw packet data from the single packet which triggered the alert.

Reporting and Analysis

Alerts are transmitted from the IPS device to the management station – such as *Configware Insite* (as well as other third party products, if required) - via SNMP traps. Trap notification is set up through the device's *Target Address* table where the administrator specifies SNMP parameters and selects which type of notification the target server will receive. In the *Community Table*, the administrator can designate that specific users have access to the traps.

Security events are also logged to an all-purpose cyclic Log File. The device's Log File can be obtained at any time, but is of limited size. When the number of entries is beyond the permitted limit, the oldest entries are overwritten. Notifications are raised when the file is 80 per cent utilised, and 100 per cent utilised.

The *Attack Reports Desktop* allows the administrator to access all the reporting options. Attack reports provide attack performance and impact on the network in a graphical layout. Historical reports show attack activity over time, and it is possible to view the top ten attacks on the system and how they change over a specified period.

Attack reports are created using information selected from security event logs. Radware provides a set of predefined reports to examine the type of attacks affecting the protected network, and their volume, bandwidth or severity. It is also possible to select individual bars or pie-sections on the graphical reports and drill down (one level only) to view the data behind them.

The following predefined Attack Reports are available:

- ***Top Attacks -*** *Graphs the top ten attacks according to packet count per attack.*

- ***Top Attacks by Category -*** *Graphs the top ten attack groups (Intrusions, DoS, Anomalies, SYN Floods, and Anti-Scanning), calculated according to packet count per group.*
- ***Top Attack Targets -*** *Graphs the top ten attack target destinations per IP Address.*
- ***Top Attack Sources -*** *Graphs the top attacks according to attack sources per IP Address.*
- ***Top Attack Targets Bandwidth -*** *Graphs the top ten attacks by bandwidth consumption.*
- ***Number of Attacks Over Time -*** *Graphs the changes in total number of attacks over a specified time period.*
- ***Attacks by Severity -*** *Graphs the attacks ranked by severity of risk: High/Medium/Low*
- ***Top Attacks by Module -*** *Graphs the top ten policies in use, ranked by packet volume per policy, per module (Intrusions, DoS, Anomalies, SYN Floods, and Anti-Scanning).*

Along with predefined reports that provide pre-configured types of network analysis, it is possible to set filtering parameters to create custom reports for viewing attack activity. It is possible to create graphs for high-level views or more detailed drill-down views of network attacks, though, once again, it is not possible to save complex filter combinations as complete custom reports.

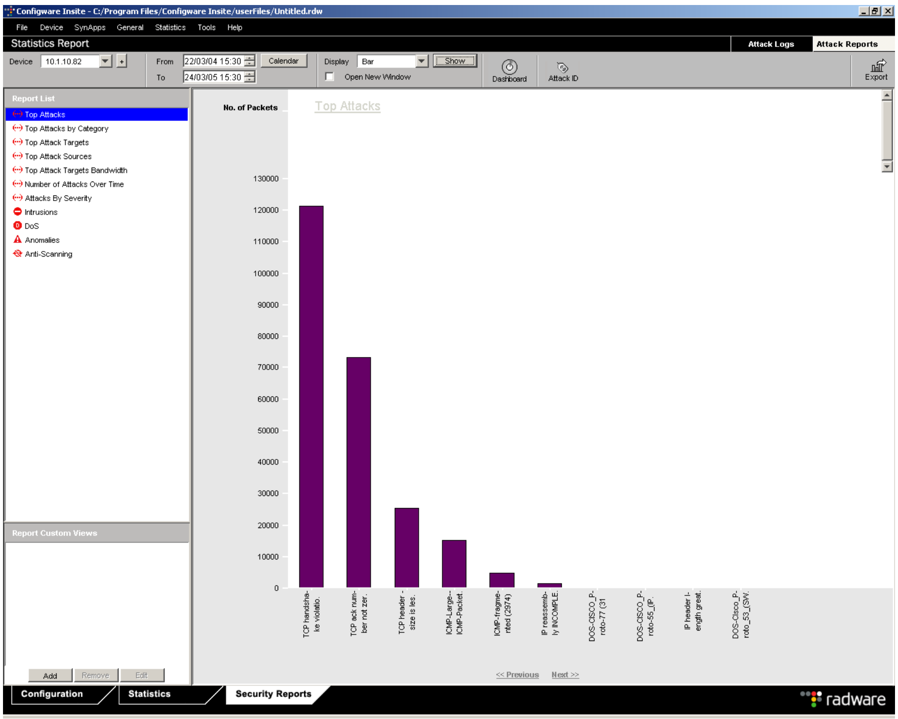

Figure 10 - DefensePro: Reports

Executive Security reports can be generated and exported in HTML format. These reports allow the generation of reports that are composed of more than one graph.

The Executive Report can include one or more of the following reports, all displayed as pie charts:

- ***Top 10 Attacks -*** *Displays the top 10 attacks with packet count*

- ***Top 10 Attack Sources -*** *Display the top 10 attack sources with packet count*
- ***Top 10 Attack Source and Destination -*** *Displays the top ten attack source and destination addresses with packet count*
- ***Top Attack Destinations -*** *Display the top 10 attacked destinations with packet count*
- ***Attacks by Category -*** *Displays the top 10 attacks including their Category (Intrusions, Anomalies, etc.) with packet count.*
- ***Attacks by Risk -*** *Displays the top 10 attacks including their Risk and packet count.*

Apart from the usual reports and graphs, Configware Insite provides a useful *Security Dashboard* feature providing a real time attack view displaying the most recent attack activity in the network.

The Security Dashboard also provides extracts of key Attack Reports and the immediate status of specific attacks. These reports graph the most intensive (top) attacks by packet volume, and the Dashboard can be refreshed automatically at user-defined intervals.

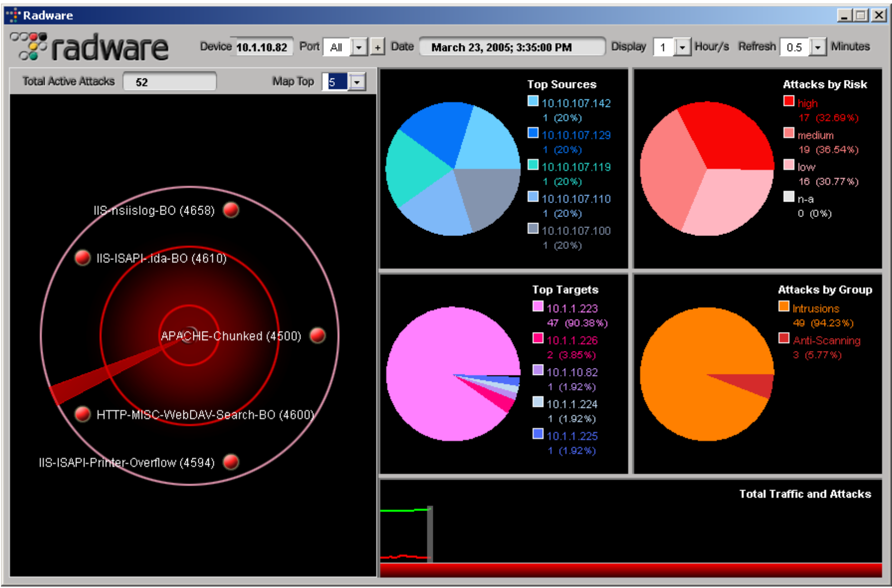

Figure 11 - DefensePro: Dashboard

The Dashboard has two panels. To the left is the Top Security Attacks Radar, which displays the most intensive attacks currently in the system, whilst to the right are four graphs which graph the top attacks in the network and their severity.

These four graphs provide a more comprehensive picture of real-time attacks to the system by mapping the following:

- ***Total Number of Attacks -*** *Shows the current total number of attacks and the total for the display period.*
- ***Attacks By Severity -*** *Breakdown of attacks in the display period by severity: High, Medium, Low.*
- ***Top Attack Targets –*** *IP addresses of the top five attack targets for the display period.*
- ***Top Attack Sources -*** *IP addresses of the top five attack sources for the display period.*

Verdict

Performance

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

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

The DefensePro is rated by Radware at 3Gbps, and was tested to 1Gbps in this test. It turned in a good performance in almost all the tests, indicating that it can easily handle 1Gbps (and more) of normal network traffic.

DefensePro detected and, more importantly, blocked all attacks even when subjected to extreme loads, and under all other load conditions it performed well. At the more extreme loads (approaching 1Gbps at the higher connection rates), the device did exhibit slightly higher HTTP response times, and the occasional failed TCP connection. We also noted an inability to process the full 20,000 connections per second at 1Gbps, or the full 10,000 *delayed* connections per second at 1Gbps (see Test Results section for full details). Despite this, we would rate DefensePro as a true 1Gbps device.

DefensePro’s basic latency figures were excellent across the board under all traffic loads, ranging from 117µs with 250Mbps of 256 byte packets, to 201µs with 1Gbps of 1000 byte packets.

Behaviour throughout the tests with no background traffic was consistent and predictable, with minimal increases as additional network load was applied from 250Mbps to 1Gbps. There were also minor increases when placing the device under a half load of 500Mbps of HTTP traffic, rising from 117µs to 160µs with 256 byte packets, 140µs to 183µs with 550 byte packets, and from 179µs to 219µs with 1000 byte packets.

100Mbps of SYN flood traffic was barely registered by the device, resulting in negligible (less than 10µs) increases in latency compared with the base figures. HTTP response times were very good overall and, once again, the addition of a 100Mbps SYN flood attack had a negligible effect on the performance.

Overall, latency figures were considered to be excellent for a device of this type under all load conditions and packet sizes. Clearly this device can be placed anywhere on the corporate network - from the perimeter to a heavily-loaded high-speed backbone - without significantly impacting overall network performance in any way.

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

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

Security Effectiveness

We installed one sensor with the latest updates, and enabled all signatures except for *Protocol Anomalies* and the *Archive* group (i.e. retired signatures).

Signature recognition was improved to 88 per cent following the application of a signature update after 24 hours, increased from a barely adequate 71 per cent out of the box. Blocking performance was one per cent higher throughout, due to one exploit being consistently blocked without an alert being raised. We consider this level of performance to be only just acceptable.

Performance in our “false negative” tests was poor out of the box, and although it improved following the signature update, there were still five misses out of the 14 test cases. This could indicate that many signatures are written for specific exploits rather than for the underlying vulnerability – perhaps an over-reliance on basic pattern matching rather than protocol decode.

A major concern in deploying an IPS is the blocking of legitimate traffic. All the tests passed successfully upon signature file update, although DefensePro turned in a less than perfect performance out of the box, failing in 5 out of 17 test cases.

Resistance to known evasion techniques was very good, with the DefensePro achieving a clean sweep across the board in most of our evasion tests. *Fragroute* and *Whisker* both failed to deceive the device into ignoring valid attacks, and many of the attempts were decoded accurately. Of the miscellaneous evasion techniques, changing ports on Trojan programs and using RPC fragmentation both proved troublesome.

Out of the box, Radware claims that DefensePro can handle approximately 1,100,000 open connections with IP and TCP reassembly disabled (the default is 800,000). We did not attempt to verify this in our tests since we believe such anti-evasion features should always be enabled. We were able to verify up to 500,000 connections without tuning, but it was not possible to increase this to 1 million, since the device did not have enough memory to support this level of open connections with IP and TCP reassembly enabled. We also felt that the session ageing time was too low, causing state to be lost too early.

Stateless “exploits” are not alerted upon (this is correct behaviour in order to be resistant to *Stick* and *Snot* tools) and mid-flows are blocked by default (a mid-flow violation alert is raised). It is, however, possible to configure the device to allow mid-flows, and there is a configurable “grace period” where they are not enforced following a power-cycle to prevent blocking of legitimate traffic should the device come on-line in mid session.

Usability

With multiple methods of managing the device DefensePro offers the administrator a choice of how to approach his management tasks.

The system offers a full-featured, text-based command-line interface, a two-tier system using a browser-based interface, and a three-tier system using a Java-based interface.

We found the dashboard with "radar" display to be both attractive and useful, given that it also supports limited drill-down capabilities.

DefensePro is straightforward to install and configure and manage on a daily basis. Policy creation is very flexible and powerful, though not always as intuitive as we might like.

We would like to see some additional refinement of the policy editor, notably a search function for individual signatures and groups of signatures, and the ability to enable and disable individual signatures as well as groups once they have been added to a profile.

The biggest issue, however, is the inability to manage multiple devices simultaneously. Although all devices can be managed from a central console, it is necessary to connect to each individual device in order to deploy policies. It should be possible to create a single policy for subsequent distribution to multiple DefensePro devices - this is a serous omission for an enterprise-class IPS system.

Alert handling is adequate, but is missing the ability to select individual elements of the alerts and drill down or up to further analyse the attacks. For example, highlighting the source IP address on one alert, it would be useful to be able to right click, and generate a view of all attacks from that same source IP address. Radware is rolling out a new CWIS version that will support such drill-down capabilities.

It would also be useful to be able to go straight into the appropriate signature within a policy from an alert to disable the signature or fine tune its configuration parameters.

Reporting is adequate, with high-level graphical views offering the ability to drill down to the detail beneath, although we would prefer to be able to drill down more than one level.

Contact Details

Company name: Radware

E-mail: info_int@radware.com

Internet: www.radware.com

Address:
22 Raoul Wallenberg Street
Tel Aviv 69710
Israel

Tel: +972 3 766 8666
Fax: +972-3-766 8655

APPENDIX A – TEST RESULTS

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

The Test Environment

The network is 100/1000Mbit Ethernet with CAT 5e cabling and Cisco 6500-Series switches (these have a mix of fibre and copper Gigabit interfaces). All devices are expected to be provided as appliances - if software-only, the supplier pre-installs the software on the recommended hardware platform. The sensor is configured as a perimeter device during testing (i.e. as if installed behind the main Internet gateway/firewall). There is no firewall protecting the target subnet.

Traffic generation equipment - such as the machines generating exploits, Spirent Avalanche and Spirent Smartbits *transmit* port - is connected to the “external” network, whilst the “receiving” equipment - such as the “target” hosts for the exploits, Spirent Reflector and Spirent Smartbits *receive* port - is connected to the internal network. The device under test is connected between two “gateway” switches - one at the edge of the external network, and one at the edge of the external network.

All “normal” network traffic, background load traffic and exploit traffic will therefore be transmitted **through** the device under test, from external to internal. The same traffic is mirrored to a single SPAN port of the external gateway switch, to which an Adtech network monitoring device is connected. The Adtech AX/4000 monitors the same mirrored traffic to ensure that the total amount of traffic never exceeds 1Gbps (which would invalidate the test run).

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

Section 1 – Detection Engine

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

Test 1.1 - Attack Recognition

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

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

- ***Test 1.1.1 -*** *Backdoors (standard ports and random ports)*
- ***Test 1.1.2 -*** *DNS/WINS*
- ***Test 1.1.3 -*** *DOS*
- ***Test 1.1.4 -*** *False negatives (common exploits which have been modified to remove or alter obvious "triggers" - this ensures that the signatures are coded for the underlying vulnerability rather than a particular exploit)*
- ***Test 1.1.5 -*** *Finger*
- ***Test 1.1.6 -*** *FTP*
- ***Test 1.1.7 -*** *HTTP*
- ***Test 1.1.8 -*** *ICMP (including unsolicited ICMP response)*
- ***Test 1.1.9 -*** *Reconnaissance*
- ***Test 1.1.10 -*** *RPC*
- ***Test 1.1.11 -*** *SSH*
- ***Test 1.1.12 -*** *Telnet*
- ***Test 1.1.13 -*** *Database*
- ***Test 1.1.14 -*** *Mail*
- ***Test 1.1.15 -*** *Voice*

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

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

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

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

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

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

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

Test 1.2 - Resistance To False Positives

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

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

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

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

Section 2 – Evasion

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

Test 2.1 - Baselines

The aim of this test is to establish that the sensor is capable of detecting and blocking a number of common basic attacks (our baseline suite) in their normal state, with no evasion techniques applied. Note that common/older attacks have been chosen deliberately for this particular test to ensure that ALL products tested have signatures in place for the evasion tests.

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

Test 2.2 - Packet Fragmentation and Stream Segmentation

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

- ***Test 2.2.1 -*** *IP fragmentation - ordered 8 byte fragments*
- ***Test 2.2.2 -*** *IP fragmentation - ordered 24 byte fragments*
- ***Test 2.2.3 -*** *IP fragmentation - out of order 8 byte fragments*
- ***Test 2.2.4 -*** *IP fragmentation - ordered 8 byte fragments, duplicate last packet*
- ***Test 2.2.5 -*** *IP fragmentation - out of order 8 byte fragments, duplicate last packet*
- ***Test 2.2.6 -*** *IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse*

- ***Test 2.2.7 -*** *IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new)*
- ***Test 2.2.8 -*** *IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old)*
- ***Test 2.2.9 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums*
- ***Test 2.2.10 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags*
- ***Test 2.2.11 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream*
- ***Test 2.2.12 -*** *TCP segmentation - ordered 1 byte segments, duplicate last packet*
- ***Test 2.2.13 -*** *TCP segmentation - ordered 2 byte segments, segment overlap (favour new)*
- ***Test 2.2.14 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers*
- ***Test 2.2.15 -*** *TCP segmentation - out of order 1 byte segments*
- ***Test 2.2.16 -*** *TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits*
- ***Test 2.2.17 -*** *TCP segmentation - ordered 1 byte segments, segment overlap (favour new)*
- ***Test 2.2.18 -*** *TCP segmentation - out of order 1 byte segments, PAWS elimination (interleaved dup segs with older TCP timestamp options)*
- ***Test 2.2.19 -*** *IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery*
- ***Test 2.2.20 -*** *TCP segmentation - ordered 16 byte segments, segment overlap (favour new (Unix))*

For each of the evasion techniques, we note if (i) the attempted attack is blocked successfully (the primary aim of any IPS device), (ii) the attempted attack is detected and an alert raised in **any** form, and (iii) if the exploit is successfully “decoded” to provide an accurate alert relating to the original exploit, rather than alerting purely on anomalous traffic detected as a result of the evasion technique itself.

Test 2.3 - URL Obfuscation

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

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

For each of the evasion techniques, we note if (i) the attempted attack is blocked successfully, (ii) the attempted attack is detected and an alert raised in **any** form, and (iii) if the exploit is successfully "decoded" to provide an accurate alert relating to the original exploit, rather than alerting purely on anomalous traffic detected as a result of the evasion technique itself.

Test 2.4 - Miscellaneous Evasion Techniques

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

- ***Test 2.4.1 -*** *Altering default ports/passwords for backdoors*
- ***Test 2.4.2 -*** *Inserting spaces in FTP command lines*
- ***Test 2.4.3 -*** *Inserting non-text Telnet opcodes in FTP data stream*
- ***Test 2.4.4 -*** *Polymorphic mutation (ADMmutate)*
- ***Test 2.4.5 -*** *Altering protocol and RPC PROC numbers*
- ***Test 2.4.6 -*** *RPC record fragging (MS-RPC and Sun)*
- ***Test 2.4.7 -*** *HTTP exploits to non-standard port*

For each of the evasion techniques, we note if (i) the attempted attack is blocked successfully, (ii) the attempted attack is detected and an alert raised in **any** form, and (iii) if the exploit is successfully "decoded" to provide an accurate alert relating to the original exploit, rather than alerting purely on anomalous traffic detected as a result of the evasion technique itself.

Section 3 – Stateful Operation

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

Test 3.1 - Stateless Attack Replay (Mid-Flows)

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

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

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

In order to receive a "PASS" in this test, no alerts should be raised for any of the actual exploits (although "mid-flow" alerts are permitted).

However, each packet should be blocked if possible since it represents a “broken” or “incomplete” session.

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

Test 3.2 - Simultaneous Open Connections (default settings)

This test determines whether the sensor is capable of preserving state across increasing numbers of open connections, as well as continuing to detect and block new exploits when the state tables are filled. It also attempts to determine whether or not the sensor will block legitimate traffic once state tables are filled. This test is run using the default sensor settings (no tuning of sensor parameters).

A legitimate HTTP session is opened and the first packet of a two-packet exploit is transmitted. The Spirent Avalanche (on the “external” interface of the sensor) then opens various numbers of TCP sessions from 10,000 to 1,000,000 (one million) with the Spirent Reflector (on the “internal” interface of the sensor). The initial HTTP session is then completed with the second half of the exploit and the session is closed. If the sensor is still maintaining state on the first session established, the exploit will be recorded. If the state tables have been exhausted, the exploit string will be seen as a non-stateful attack, and will thus be ignored.

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

At each step, we ensure that the sensor is still capable of detecting and blocking freshly-launched exploits once all the connections are open, as well as confirming that the device does not block legitimate traffic (perhaps as a result of state tables filling up). We then launch further exploits whilst the Avalanche/Reflector devices “churn” connections at the maximum level set, ensuring that the sensor is still capable of detecting and blocking freshly-launched exploits as old connections are torn down and new ones recreated constantly.

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

Test 3.3 - Simultaneous Open Connections (after tuning)

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

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

Section 4 – Detection/Blocking Performance Under Load

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

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

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

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

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

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

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

Test 4.1 - UDP Traffic To Random Valid Ports

This test uses UDP packets of varying sizes generated by a **Smartbits SMB6000** with LAN-3301A 10/100/1000Mbps **TeraMetrics** cards installed.

A constant stream of the appropriate mix of packets - with variable source IP addresses and ports transmitting to a single fixed IP address/port - is transmitted through the sensor (bi-directionally, maximum of 1Gbps).

Each packet contains dummy data, and is targeted at a valid port on a valid IP address on the target subnet. The percentage load and packets per second (pps) figures are verified by the Adtech Gigabit network monitoring tool before each test begins. Multiple tests are run and averages taken where necessary.

This traffic does not attempt to simulate any form of “real world” network condition. The aim of this test is purely to determine the raw packet processing capability of the sensor, and its effectiveness at passing “useless” packets quickly in order to pass potential attack packets to the detection engine. The range of packet sizes has been selected to mirror the maximum, minimum and average packet sizes used in our HTTP stress tests.

- ***Test 4.1.1 - 256 byte packets - maximum 453,000 packets per second:*** *This test is roughly equivalent to a 40,000 connections per second test in our HTTP stress tests (in terms of packet size and packets per second rate), and has been included to provide an indication of the packet processing performance under the most extreme conditions for most devices - it is unlikely that any real-life network will ever see network loads of over 450,000 256-byte packets per second unless under severe DOS conditions. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic.*
- ***Test 4.1.2 - 550 byte packets - maximum 220,000 packets per second:*** *This test has been included to provide a comparison with our “real world” packet mixes, since the average packet size is similar. No sessions are created during this test and there is very little for the detection engine to do in the way of protocol analysis. This test provides a reasonable indication of the ability of a device to process packets from the wire on an “average” network, and we would expect all products to demonstrate good performance levels. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic.*
- ***Test 4.1.3 - 1000 byte packets - maximum 122,000 packets per second:*** *This test is the complete opposite of the 256 byte packet test, in that we would expect every single product to be capable of returning 100 per cent detection rates across the board when using only 1000 byte packets. We have included this test mainly to demonstrate how easy it is to achieve good results using large packets – beware of test results that* ***only*** *quote performance figures using similar (or larger) packet sizes. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic.*

Test 4.2 - HTTP “Maximum Stress” Traffic With No Transaction Delays

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

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

By creating genuine session-based traffic with varying session lengths, the sensor is forced to track valid sessions, thus ensuring a higher workload than for simple packet-based background traffic. This provides a test environment that is as close to “real world” as it is possible to achieve in a lab environment, whilst ensuring absolute accuracy and repeatability.

The aim of this test is to stress the HTTP detection engine and determine how the sensor copes with detecting and blocking exploits under network loads of varying average packet size and varying connections per second.

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

- ***Test 4.2.1 -*** *Max 2,500 new connections per second - average packet size 1000 bytes - maximum 120,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With relatively low connection rates and large packet sizes, we expect all sensors to achieve 100% blocking rates throughout this test.*
- ***Test 4.2.2 -*** *Max 5,000 new connections per second - average packet size 540 bytes - maximum 225,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With average connection rates average packet sizes, this is a good approximation of a real-world production network, and we expect all sensors to perform well in this test.*
- ***Test 4.2.3 -*** *Max 10,000 new connections per second - average packet size 440 bytes - maximum 275,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With average packet sizes coupled with very high connection rates, this is a strenuous test for any sensor, and represents a very heavily used production network.*
- ***Test 4.2.4 -*** *Max 20,000 new connections per second - average packet size 360 bytes - maximum 320,000 packets per second. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With small packet sizes and extremely high connection rates this is an extreme test for any sensor. Not many sensors will perform well at all levels of this test.*

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

This test is identical to Test 4.2 except that we introduce a 10 second delay in the server response for each transaction. This has the effect of maintaining a high number of open connections throughout the test, thus forcing the sensor to utilise additional resources to track those connections.

- ***Test 4.3.1 -*** *Max 5,000 new connections per second - average packet size 540 bytes - maximum 225,000 packets per second - 10 second transaction delay - maximum 50,000 open connections. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With average connection rates average packet sizes, this is a good approximation of a real-world production network, and we expect all sensors to perform well in this test.*
- ***Test 4.3.2 -*** *Max 10,000 new connections per second - average packet size 440 bytes - maximum 275,000 packets per second - 10 second transaction delay - maximum 100,000 open connections. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With average packet sizes coupled with very high connection rates, this is a strenuous test for any sensor, and represents a very heavily used production network.*

Test 4.4 - Protocol Mix Traffic

Whereas 4.2 and 4.3 provide a pure HTTP environment with varying connection rates and average packet sizes, the aim of this test is to simulate more of a “real world” environment by introducing additional protocols whilst still maintaining a precisely repeatable and consistent background traffic load (something rarely seen in a real world environment).

The result is a background traffic load that, whilst less stressful than previous tests, is closer to what may be found on a heavily-utilised “normal” production network.

- ***Test 4.4.1 -*** *72% HTTP traffic (540 byte packets) + 20% FTP traffic + 6% UDP traffic (256 byte packets). Max 4000 new connections per second - average packet size 540 bytes - maximum 215,000 packets per second - maximum 750 open connections. Repeated with 250Mbps, 500Mbps, 750Mbps and 1000Mbps of background traffic. With lower connection rates, average packets sizes and a common protocol mix, this is a good approximation of a heavily-used production network, and we expect all sensors to perform well throughout this test.*

Test 4.5 - “Real World” Traffic

This is as close as it is possible to come to a true “real world” environment under lab conditions. For this test we eliminate the Reflector device and substitute an IIS Web server installed on a dual-Xeon server with Gigabit interface and 4GB RAM. This server holds a copy of The NSS Group Web site, and is capable of handling a full 1Gbps of traffic. We then capture a typical client browsing session on the NSS Group Web site, accessing a mixture of menu pages, lengthy text-based reports and multiple graphical images (screen shots) and have Avalanche replay multiple identical sessions from up to **20 new users per second**.

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

- ***Test 4.5.1 - Pure HTTP Traffic (simulated browsing session on NSS Web site):*** *Max 4700 new connections per second - 20 new users per second - average packet size 560 bytes - maximum 210,000 packets per second.*

*Repeated with 250Mbps, 500Mbps, 750Mbps and 950Mbps of background traffic. With genuine server responses to genuine **browser** sessions consisting **of multiple transactions per** session, this is a typical "real world" background load, albeit pure HTTP. Although the Web server and the network are extremely busy at the higher traffic loads, the "normal" connection rates and packet sizes should enable most sensors to perform well at all load levels in this test.*

- ***Test 4.5.2 - Protocol Mix (72% HTTP traffic (simulated browsing sessions as 4.5.1)) + 20% FTP traffic + 6% UDP traffic (256 byte packets)):*** *Max 3700 new connections per second - average packet size 560 bytes - maximum 205,000 packets per second - maximum 1,500 open connections.*

  *Repeated with 250Mbps, 500Mbps, 750Mbps and 950Mbps of background traffic. With genuine server responses to genuine browser sessions consisting of multiple **transactions** per session, mixed with FTP and UDP traffic, this is a typical "real world" background load. Although the Web server and the network are extremely busy at the higher traffic loads, the "normal" connection rates and packet sizes should enable most sensors to perform well at all load levels in this test.*

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

Section 5 – Latency & User Response Times

The aim of this section is to determine the effect the sensor has on the traffic passing through it under various load conditions.

Should a device impose a high degree of latency on the packets passing through it, a network or security administrator would need to think carefully about how many devices could be installed in a single data path before user response times became unacceptable or the combination of devices caused excessive timeouts. We also determine the effect of high levels of normal HTTP traffic and a basic DOS attack on the average latency and user response times.

Test 5.1 - Latency

We use Spirent SmartFlow software and The Smartbits SMB6000 with Gigabit TeraMetrics cards to create multiple traffic flows through the appliance and measure the basic throughput, packet loss, and latency through the sensor. This test - whilst not indicative of real-life network traffic - provides an indication of how much the sensor affects the traffic flow through it. This data is particularly useful for network administrators who need to gauge the effect of any form of in-line device which is likely to be placed at critical points within the corporate network.

SmartFlow runs through several iterations of the test varying the traffic load from 250Mbps to 1Gbps bi-directionally (or up to the maximum rated throughput of the device should this be less than 1Gbps) in steps of 250Mbps. This is repeated for a range of packet sizes (256 bytes, 550 bytes and 1000 bytes) of UDP traffic with variable IP addresses and ports. At each iteration of the test, SmartFlow records the number of packets dropped, together with average and maximum latency.

- ***Test 5.1.1 - Latency With No Background Traffic:*** *SmartFlow traffic is passed across the infrastructure switches and through the device (the latency of the basic infrastructure is known and is constant throughout the tests). The packet loss and average latency are recorded at each packet size and each load level from 250Mbps to 1Gbps (in 250Mbps steps).*
- ***Test 5.1.2 - Latency With Background Traffic Load:*** *The Avalanche and Reflector are configured to generate a fixed amount of background HTTP traffic through the sensor (up to 50 per cent of the maximum rated bandwidth of the device under test - maximum 500Mbps - maximum 2,500 new connections per second - average packet size 540 bytes - maximum 112,500 packets per second).*

  A 250Mbps bi-directional load of SmartFlow traffic at various packet sizes (256 bytes, 540 bytes and 1000 bytes) is then passed across the infrastructure switches and through the device and the packet loss and average latency are recorded.
- ***Test 5.1.3 - Latency When Under Attack:*** *The Spirent WebSuite software is used to generate a fixed load of DOS/DDOS traffic of 10 per cent of the maximum rated bandwidth of the device under test (maximum 100Mbps). A 250Mbps bi-directional load of SmartFlow traffic at various packet sizes (256 bytes, 540 bytes and 1000 bytes) is then passed across the infrastructure switches and through the device and the packet loss and average latency are recorded. The device should be configured to detect/block/mitigate the DOS attack by the most efficient method available.*

Test 5.2 - User Response Times

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

- ***Test 5.2.1 - Web Response With No Background Traffic:*** *The Avalanche and Reflector are configured to generate HTTP traffic through the sensor (up to 50 per cent of the maximum rated bandwidth of the device under test - maximum 500Mbps - maximum 2,500 new connections per second - average packet size 540 bytes - maximum 112,500 packets per second).*

  The minimum, maximum and average page response times and number of failed connections are recorded by Avalanche to provide an indication of the expected response times under normal traffic conditions.
- ***Test 5.2.2 - Web Response When Under Attack:*** *The Avalanche and Reflector are configured to generate HTTP traffic through the sensor as for Test 5.2.1. The Spirent WebSuite software is then used to generate DOS/DDOS traffic up to 10 per cent of the maximum rated bandwidth of the device under test (maximum 100Mbps).*

  The minimum, maximum and average page response times and number of failed connections are recorded by Avalanche to provide an indication of the expected response times when the device is under attack.

Section 6 – Stability & Reliability

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

- ***Test 6.1.1 - Blocking Under Extended Attack:*** *For this test, we expose the external interface of the device to a constant stream of alerts over an extended period of time. The device is configured to block and alert, and thus this test provides an indication the effectiveness of both the blocking and alert handling mechanisms. A continuous stream of exploits mixed with some legitimate sessions is transmitted through the device at a maximum of 100Mbps (max 50,000 packets per second, average packet sizes in the range of 120-350 bytes) for 8 hours with no additional background traffic. This is not intended as a stress test in terms of traffic load - merely a reliability test in terms of consistency of blocking performance.*

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

- ***Test 6.1.2 - Passing Legitimate Traffic Under Extended Attack:*** *This test is identical to 6.1.1, where we expose the external interface of the device to a constant stream of alerts over an extended period of time. The device is expected to remain operational and stable throughout this test, and to pass 100 per cent of legitimate traffic. Results are presented as a simple PASS/FAIL. If any legitimate traffic is blocked - caused by either the volume of traffic* ***or*** *the sensor failing closed for any reason - this will result in a FAIL.*

- ***Test 6.1.3 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC:*** *This test attempts to stress the protocol stack of the device under test by exposing it to traffic from the ISIC test tool. The ISIC test tool host is connected directly to the external interface of the sensor, and the ISIC target directly to the internal interface. ISIC traffic is transmitted through the sensor (without passing through any other network equipment) and the effects noted. Traffic load is a maximum of 350Mbps and 60,000 packets per second (average packet size is 690 bytes). Results are presented as a simple PASS/FAIL - the device is expected to remain operational and capable of detecting and blocking exploits throughout the test to attain a PASS.*

Section 7 – Management and Configuration

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

Test 7.1 - Management Port

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

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

- ***Test 7.1.1 - Open ports:*** *We will scan the open ports and active services on the management interface and report on known vulnerabilities.*
- ***Test 7.1.2 - ISIC/ESIC/TCPSIC/UDPSIC/ICMPSIC:*** *This test attempts to stress the protocol stack of the management interface of the device under test by exposing it to traffic from the ISIC test tool. The ISIC test tool host is connected directly to the management interface of the IPS sensor, and that interface is also the target. ISIC traffic is transmitted to the management interface of the IPS device (without passing through any other network equipment) and the effects noted.*

  Traffic load is a maximum of 350Mbps and 60,000 packets per second (average packet size is 690 bytes). Results are presented as a simple PASS/FAIL - the device is expected to remain (a) operational and capable of detecting and blocking exploits, and (b) capable of communicating in both directions with the management server/console throughout the test to attain a PASS.
- ***Test 7.1.3 -*** *We note whether the ISIC attacks themselves are detected by the sensor even though targeted at the management port.*

Radware DefensePro-3000 V2.40 Test Results

Section 1 - Detection Engine

Test 1.1 – Attack Recognition	Attacks	Default ARRD	Default ARRB	Custom ARRD	Custom ARRB
Test 1.1.1 - Backdoors	7	6	6	7	7
Test 1.1.2 - WINS/DNS	3	3	3	3	3
Test 1.1.3 - DOS	10	7	8	9	10
Test 1.1.4 - False negatives (modified exploits)	14	4	4	9	9
Test 1.1.5 - Finger	4	4	4	4	4
Test 1.1.6 - FTP	5	3	3	5	5
Test 1.1.7 - HTTP	43	32	32	39	39
Test 1.1.8 - ICMP	2	2	2	2	2
Test 1.1.9 - Reconnaissance	8	8	8	8	8
Test 1.1.10 - RPC	9	7	7	7	7
Test 1.1.11 - SSH	1	0	0	1	1
Test 1.1.12 - Telnet	1	0	0	1	1
Test 1.1.13 - Database	1	1	1	1	1
Test 1.1.14 - Mail	1	1	1	1	1
Test 1.1.15 - Voice	1	0	0	0	0
Total	**110**	**78 / 110**	**79 / 110**	**97 / 110**	**98 / 110**
		71%	**72%**	**88%**	**89%**

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	FAIL	PASS
Test 1.2.11 - SMTP e-mail transfer with “suspicious” DLL attachment	FAIL	PASS
Test 1.2.12 - SMTP e-mail transfer with “suspicious” Web page attachment	FAIL	PASS
Test 1.2.13 - SNMP V3 packet with invalid parameter	PASS	PASS
Test 1.2.14 - Fake DNS /bin/sh buffer overflow	FAIL	PASS
Test 1.2.15 - Inter-firewall communication traffic	PASS	PASS
Test 1.2.16 - Fake SQL Slammer traffic	FAIL	PASS
Test 1.2.17 - File copy of GIF file (contains bytes which look like NOP sled)	PASS	PASS
Total Passed	**12 / 17**	**17 / 17**

Section 2 - IPS Evasion

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

Test 2.2 – Packet Fragmentation/Stream Segmentation	Detected?	Decoded?	Blocked?
Test 2.2.1 - IP fragmentation - ordered 8 byte fragments	YES	YES	YES
Test 2.2.2 - IP fragmentation - ordered 24 byte fragments	YES	YES	YES
Test 2.2.3 - IP fragmentation - out of order 8 byte fragments	YES	YES	YES
Test 2.2.4 - IP fragmentation - ordered 8 byte fragments, duplicate last packet	YES	YES	YES
Test 2.2.5 - IP fragmentation - out of order 8 byte fragments, duplicate last packet	YES	YES	YES
Test 2.2.6 - IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse	YES	YES	YES
Test 2.2.7 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new)	YES	YES	YES
Test 2.2.8 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old)	YES	YES	YES
Test 2.2.9 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums	YES	NO	YES[1]
Test 2.2.10 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags	YES	NO	YES[1]
Test 2.2.11 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence nos. mid-stream	YES	YES	YES
Test 2.2.12 - TCP segmentation - ordered 1 byte segments, duplicate last packet	YES	YES	YES
Test 2.2.13 - TCP segmentation - ordered 2 byte segments, segment overlap (favour new)	YES	YES	YES
Test 2.2.14 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers	YES	NO	YES[1]
Test 2.2.15 - TCP segmentation - out of order 1 byte segments	YES	NO	YES[1]
Test 2.2.16 - TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits	YES	NO	YES[1]
Test 2.2.17 - TCP segmentation - ordered 1 byte segments, segment overlap (favour new)	YES	NO	YES[1]
Test 2.2.18 - TCP segmentation - out of order 1 byte segments, PAWS elimination (interleaved dup segments with older TCP timestamp options)	YES	NO	YES[1]
Test 2.2.19 - IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery	YES	YES	YES[2]
Test 2.2.20 - TCP segmentation - ordered 16 byte segments, segment overlap (favour new (Unix))	YES	YES	YES
Total	**20 / 20**	**13 / 20**	**20 / 20**

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

Test 2.4 – Miscellaneous Obfuscation Techniques	Detected?	Decoded?	Blocked?
Test 2.4.1 - Altering default ports	NO	NO	NO
Test 2.4.2 - Inserting spaces in FTP command lines	YES	YES	YES
Test 2.4.3 - Inserting non-text Telnet opcodes in FTP data stream	YES	YES	YES
Test 2.4.4 - Polymorphic mutation (ADMmutate)	YES	YES	YES
Test 2.4.5 - Altering protocol and RPC PROC numbers	NO	NO	NO
Test 2.4.6 - RPC record fragging (MS-RPC and Sun)	NO[3]	NO[3]	NO[3]
Test 2.4.7 - HTTP exploits to port <> 80	YES[4]	YES[4]	YES[4]
Total	**4 / 7**	**4 / 7**	**4 / 7**

Section 3 - Stateful Operation

Test 3.1 – Stateless Attack Replay	Alert?	Blocked?	Pass/Fail
Test 3.1.1 - Stateless Web exploits	NO	YES[5]	PASS
Test 3.1.2 - Stateless FTP exploits	NO	YES[5]	PASS

Test 3.2 – Simultaneous Open Connections (default settings)							
Number of open connections	10,000	25,000	50,000	100,000	250,000	500,000	1,000,000
Test 3.2.1 - Attack Detection	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Test 3.2.2 - Attack Blocking	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Test 3.2.3 - State Preservation	PASS	PASS	PASS	PASS	PASS	PASS	FAIL
Test 3.2.4 - Legitimate traffic blocking	PASS	PASS	PASS	PASS	PASS	PASS	FAIL

Test 3.3 – Simultaneous Open Connections (after tuning)[5]							
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 - Attack Blocking	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Test 3.3.3 - State Preservation	PASS	PASS	PASS	PASS	PASS	PASS	FAIL
Test 3.3.4 - Legitimate traffic blocking	PASS	PASS	PASS	PASS	PASS	PASS	FAIL

Section 4 - Detection/Blocking Performance Under Load

Test 4.1 – UDP traffic to random valid ports		250Mbps	500Mbps	750Mbps	1Gbps	Max
Test 4.1.1 - 256 byte packet test - max 453,000pps	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	
Test 4.1.2 - 550 byte packet test - max 220,000pps	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	
Test 4.1.3 - 1514 byte packet test - max 122,000pps	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	

Test 4.2 – HTTP "maximum stress" traffic with no transaction delays		250Mbps	500Mbps	750Mbps	1Gbps	Max
Test 4.2.1 - Max 2500 connections per second - ave packet size 1000 bytes - max 120,000 packets per second	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	
Test 4.2.2 - Max 5000 connections per second - ave packet size 540 bytes - max 225,000 packets per second	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	
Test 4.2.3 - Max 10000 connections per second - ave packet size 440 bytes - max 275,000 packets per second	Detected	100%	100%	100%	100%[6]	1Gbps
	Blocked	100%	100%	100%	100%[6]	
Test 4.2.4 - Max 20000 connections per second - ave packet size 360 bytes - max 320,000 packets per second	Detected	100%	100%	100%	100%[7]	940Mbps
	Blocked	100%	100%	100%	100%[7]	

Test 4.3 – HTTP "maximum stress" traffic with transaction delays		250Mbps	500Mbps	750Mbps	1Gbps	Max
Test 4.3.1 - Max 5000 connections per second - ave packet size 540 bytes - max 225,000 packets per second - 10 sec delay - max 50,000 open connections	Detected	100%	100%	100%	100%[6]	1Gbps
	Blocked	100%	100%	100%	100%[6]	
Test 4.3.2 - Max 10000 connections per second - ave packet size 440 bytes - max 275,000 packets per second - 10 sec delay - max 100,000 open connections	Detected	100%	100%	100%	N/A[8]	750Mbps
	Blocked	100%	100%	100%	N/A[8]	

Test 4.4 – Protocol mix		250Mbps	500Mbps	750Mbps	1Gbps	Max
Test 4.4.1 - 72% HTTP (540 byte packets) + 20% FTP + 6% UDP (256 byte packets). Max 4000 connections per second - ave packet size 540 bytes - max 215,000 packets per second - max 750 open connections	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	

Test 4.5 – Real World traffic		250Mbps	500Mbps	750Mbps	1Gbps	Max
Test 4.5.1 - Pure HTTP (simulated browsing session on NSS Web site). Max 4700 connections per second - 20 new users per second - ave packet size 560 bytes - max 210,000 packets per second	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	
Test 4.5.2 - Protocol mix - 72% HTTP (simulated browsing sessions as 2.5.1) + 20% FTP + 6% UDP (256 byte packets). Max 3700 connections per second - ave packet size 560 bytes - max 205,000 packets per second - max 1,500 open connections	Detected	100%	100%	100%	100%	1Gbps
	Blocked	100%	100%	100%	100%	

Section 5 - Latency & User Response Times

Test 5.1 – Latency	Packet Size	250Mbps	500Mbps	750Mbps	1Gbps
Test 5.1.1 Average latency (μs) with no background traffic	256	116.75	120.44	132.42	199.13[9]
	550	139.51	145.78	153.75	154.05
	1000	178.56	187.51	193.45	201.14
Test 5.1.2 Average latency (μs) with background traffic (500Mbps HTTP traffic, max 2500 connections per second - ave packet size 540 bytes - max 112,500 packets per second)	256	159.87			
	550	182.82			
	1000	218.97			
Test 5.1.3 Average latency (μs) when under attack (100Mbps SYN flood)	256	125.51			
	550	148.52			
	1000	184.46			

Test 5.2 – User Response Times	Attempted Trans	Failed Trans	Min Page Response	Max Page Response	Ave Page Response
Test 5.2.1 - Web page response (ms) with no background traffic (500Mbps HTTP traffic, max 2500 connections per sec - ave packet size 540 bytes - max 112,500 packets per sec)	1556320	0	200	255	206
Test 5.2.2 - Web page response (ms) when under attack (500Mbps HTTP traffic, max 2500 connections per sec - ave packet size 540 bytes - max 112,500 packets per sec PLUS 100Mbps SYN flood)	1556393	0	201	257	206

Section 6 - Stability & Reliability

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

Section 7 - Management Interface

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

Notes:

1. Default setting is "*forward*" for this anomaly - set via CLI only. Needs to be set to "*drop*" in order to avoid evasion ("*drop*" will be the default in future releases).
2. Always blocked, regardless of configuration
3. Cannot handle MS-RPC evasion. Sun-RPC evasion handling depends on signature.
4. Requires configuration to set HTTP ports
5. Alerts on mid-flow violation. It is possible to configure the device to allow mid-flows
6. Noted slightly higher response times at this level with some occasional small packet loss and very small HTTP transaction failure. Test was stable, however, hovering around the maximum throughput requested
7. This throughput was achieved at around 940Mbps - limit was approx 18,000 connections per second (300,000 packets per second)

8. Not possible to run this test - maximum connections per second exceeded
9. This latency was achieved at 950Mbps due to slight packet loss at 1Gbps (1-4 packets per 10 million)

Section 1: Detection Engine

We installed one sensor with the latest updates, and enabled all signatures except for *Protocol Anomalies* and the *Archive* group (i.e. retired signatures). The following adjustments were also required:

- *IP reassembly is not enabled by default – this was enabled for the test*
- *100 second timeout on aged sessions by default – increased to 180 seconds*
- *SYN flood threshold was modified*

Out of the box, signature recognition was just adequate at 71 per cent, and was improved to 88 per cent following the application of a signature update after 24 hours. Blocking performance was one per cent higher throughout, due to one exploit being consistently blocked without an alert being raised. We consider this level of performance to be only just acceptable.

Performance in our “false negative” tests was poor out of the box, and although it improved following the signature update, there were still five misses out of the 14 test cases. This could indicate that many signatures are written for specific exploits rather than for the underlying vulnerability – perhaps an over-reliance on basic pattern matching rather than protocol decode.

A major concern in deploying an IPS is the blocking of legitimate traffic. Unfortunately, DefensePro turned in a less than perfect performance out of the box here too, failing on 5 out of 17 test cases. These were all rectified successfully following the signature update.

Section 2: IPS Evasion

Resistance to known evasion techniques was very good, with the DefensePro achieving a clean sweep across the board in most of our evasion tests. *Fragroute* and *Whisker* both failed to deceive the device into ignoring valid attacks, and many of the attempts were decoded accurately.

Note, however, that many of the TCP segmentation techniques were detected via a generic protocol anomaly error, and the default setting out of the box was to forward this traffic. This meant that although the exploits were detected successfully, the traffic was forwarded to the protected network. The setting in question could only be modified via the CLI on the appliance, following which all the evasion attempts were blocked completely. This setting will be changed to drop traffic by default in future releases.

Of the miscellaneous evasion techniques, changing ports on Trojan programs and using RPC fragmentation both proved troublesome (DefensePro could handle Sun RPC fragmentation on certain signatures, but not MS-RPC).

Section 3: Stateful Operation

Out of the box, Radware claims that DefensePro can handle approximately 1,100,000 open connections with IP and TCP reassembly disabled (the default is 800,000). We did not attempt to verify this in our tests since we believe such anti-evasion features should always be enabled. We were able to verify up to 500,000 connections without tuning, but it was not possible to increase this to 1 million, since the device did not have enough memory to support this level of open connections with IP and TCP reassembly enabled. We also felt that the session ageing time was too low, causing state to be lost too early.

Default operation of the device is to reject new connections when the state tables are full or resources are low, and this meant that once we exceeded the connection limit the device continued to maintain state on existing connections (thus detecting our half-open exploit), but inevitably, as a consequence, began to block legitimate traffic. This behaviour is not configurable.

Stateless “exploits” are not alerted upon (this is correct behaviour in order to be resistant to *Stick* and *Snot* tools) and mid-flows are blocked by default (a mid-flow violation alert is raised). It is, however, possible to configure the device to allow mid-flows, and there is a configurable “grace period” where they are not enforced following a power-cycle to prevent blocking of legitimate traffic should the device come on-line in mid session.

Section 4: Detection/Blocking Performance Under Load

The DefensePro is rated by Radware at 3Gbps, and was tested to 1Gbps in this test. It turned in a good performance in almost all the tests, indicating that it can easily handle 1Gbps (and more) of normal network traffic.

At the more extreme loads (approaching 1Gbps at the higher connection rates), the device did exhibit slightly higher HTTP response times, and the occasional failed TCP connection. Note that we were also unable to complete some of the tests (20,000cps HTTP tests, and 10,000cps HTTP tests with delayed transactions) due to DefensePro’s inability to process that level of connections per second at 1Gbps.

However, DefensePro continued to detect and, more importantly, block all attacks even when subjected to these extreme loads, and under all other load conditions it performed well. We would therefore be happy to rate this device at 1Gbps.

Section 5: Latency & User Response Times

DefensePro’s basic latency figures were excellent across the board under all traffic loads. They ranged from 117µs with 250Mbps of 256 byte packets, to 201µs with 1Gbps of 1000 byte packets.

Behaviour throughout the tests with no background traffic was consistent and predictable, with minimal increases as additional network load was applied from 250Mbps to 1Gbps. Placing the device under a half load of 500Mbps of HTTP traffic, also caused relatively minor increases in latency, rising from 117µs to 160µs with 256 byte packets, 140µs to 183µs with 550 byte packets, and from 179µs to 219µs with 1000 byte packets.

100Mbps of SYN flood traffic was barely registered by the device, resulting in negligible (less than 10μs) increases in latency compared with the base figures. The SYN flood was mitigated completely once it had been detected (which did not take long). DefensePro uses delayed binding and SYN cookies in order to mitigate SYN floods once the detection threshold has been exceeded - the attack mitigation capabilities of the DefensePro appear to be excellent.

HTTP response times were very good overall and, once again, the addition of a 100Mbps SYN flood attack had a negligible effect on the performance.

Overall, latency figures were considered to be excellent for a device of this type under all load conditions and packet sizes. Clearly this device can be placed anywhere on the corporate network - from the perimeter to a heavily-loaded high-speed backbone - without significantly impacting overall network performance in any way.

Section 6: Stability & Reliability

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

Exposing the sensor interface to ISIC-generated traffic had no adverse effect (other than a slight increase in latency for normal traffic during the attack), and the device continued to detect and block all other exploits throughout and following the ISIC attack.

A large number of ISIC-related alerts were raised during the attack and the attack was mitigated partially. There were no residual stability problems once the attack had been terminated.

Section 7: Management Interface

Only the serial console port is open by default - all ports on the management interface are closed. Telnet, SSH, HTTP and HTTPS ports can all be enabled/disabled as required.

The extended ISIC attack against the management interface caused no problems either during or after the attack and, unlike most of the competition, DefensePro was able to detect and alert on attacks against the management interface.