McAfee Entercept 4.1

Technical Evaluation

An NSS Group Report

First published January 2004 (Version 1.0)

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

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

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

TABLE OF CONTENTS

TABLE OF FIGURES

The NSS Group

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

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

The Group consists of two wholly-owned subsidiaries :

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

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

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

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

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

Foreword

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

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

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

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

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

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

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

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

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

Bob Walder

INTRODUCTION

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

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

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

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

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

Intrusion Prevention Systems (IPS)

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

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

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

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

Host IPS (HIPS)

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

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

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

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

Network IPS (NIPS)

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

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

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

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

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

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

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

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

The NSS Intrusion Prevention Group Test

The NSS Group has conducted the first comprehensive IPS test of its kind. This exhaustive review will give readers a complete perspective of the capabilities, maturity and suitability of the products tested for their particular needs. The NSS Group established this test because IPS products are being actively deployed as a new layer in defence-in-depth security architectures.

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

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

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

Performance

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

The aim of this section is to determine the impact the Host IPS agent has on the host on which it is installed. These tests verify that the IPS does not adversely impact legitimate traffic, in addition to ensuring that the sensor is capable of detecting and blocking exploits when subjected to increasing loads of background traffic. A Host IPS that misses attacks under load can be evaded, whilst a Host IPS that adversely affects legitimate background traffic will not stay installed for long, since high application latency will create a negative user experience.

For all tests the host is a high-performance Gigabit-capable Web server running Microsoft IIS Web server with a copy of The NSS Group Web site. Web traffic is generated by the Spirent Communications Avalanche product connected to the same switch as the host in order to provide a direct 1Gbps data path between the simulated Web clients and our host under test. We realise that most “normal” Web servers would not be expected to serve up to 1Gbps of data, but the aim in this test is to ensure that the network infrastructure does not become a bottleneck.

Our tests are intended to be representative of “real world ” Web environments, and so we use a genuine captured browsing session from the NSS Group Web site during which the user accesses our home page, and then moves on to viewing a report on-line resulting in the retrieval of several very large text pages and a large number of accompanying graphics (screen shots). The user then moves on to accessing several smaller pages and menus before completing the session. This session consists of 180 URLs spread over 16 Web pages.

We also use multiple user profiles, each with different think times (60-120 seconds), and limit the traffic to around 100Mbps (an arbitrary value, but we felt that not many public facing Web servers would have Internet connections of more than 100Mbps). In choosing these URLs and bandwidth values, we have, in fact, mirrored exactly the live NSS Group Web server in every respect - including available bandwidth - and thus feel this is an acceptable “real world” scenario.

The tests are repeated with and without the Host IPS agent installed in order to determine the overall effect it has on performance. Tests are also repeated with HTTP and FTP traffic in order to determine differences in performance between different application layer protocols.

Resistance To Evasion Techniques

These tests verify that the Host 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 consists of four parts:

- ***Baselines -*** *This establishes that the IPS is capable of detecting and blocking a number of common basic attacks (our baseline suite) in their normal state, with no evasion techniques applied.*
- ***Packet Fragmentation and Stream Segmentation -*** *The baseline HTTP attacks are repeated, running them through fragroute using 19 evasion techniques.*
- ***URL Obfuscation -*** *The baseline HTTP attacks are repeated, this time applying 9 URL obfuscation techniques made popular by the Whisker Web server vulnerability scanner.*

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.

Security Effectiveness

The aim of this section is to verify that the agent is capable of detecting a wide range of common attempted intrusions accurately and successfully preventing them without impacting the normal operation of the host.

The latest signature/update pack is acquired from the vendor, and agents are deployed with **all** available signatures enabled (some audit/informational signatures may be disabled). It may also be necessary to create custom signatures to handle some of our test cases.

The following test cases are covered:

- *Test 1.1 - Critical system files*
- *Test 1.2 - Critical Registry keys*
- *Test 1.3 - Specific Registry keys*
- *Test 1.4 - IPS components*
- *Test 1.5 - Services/Daemons*
- *Test 1.6 - Audit logs*
- *Test 1.7 - Protected data files/directories*
- *Test 1.8 - Protected applications*
- *Test 1.9 - Network services*
- *Test 1.10 - Privilege escalation*
- *Test 1.11 - Buffer overflows*
- *Test 1.12 - Web server exploits*
- *Test 1.13 - Exceptions/False Positives*

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 agent and management software and perform common, day-to-day operations with the management console. Areas evaluated include installation, configuration, policy editing, alert handling, and reporting and analysis.

MCAFEE ENTERCEPT 4.1

Executive Summary

Produced by McAfee (formerly *Entercept Security Technologies*, and now part of *Network Associates Inc.*), Entercept is a Host-based IPS (HIPS) which monitors events at the operating system or application server level.

Since Entercept does not deal with network-based exploits, it is very complementary to existing solutions that deal with attacks on that level, such as firewalls and network-based IDS or IPS products. The latest version adds welcome new features such as a new licensing scheme, key backup capabilities, additional reports, OS lockdown and custom signatures, as well as numerous improvements “under the hood”.

The host-based approach ensures that there are no issues with switched networks or encrypted traffic, and the insertion of the Agent software at kernel level means that the system is capable of protecting the host against both known and unknown attacks with a relatively small impact on the host system.

The provision of Web Server and Database Agents also secures application software within an almost impregnable vault where virtually all attacks will be prevented before hitting the server. Should an attack actually get through, it is then prevented from operating outside the scope of the application server itself. The *Code Red* worm is one example – Entercept users were protected even without the IIS patch in place since the worm was rejected at the HTTP layer before it could deliver its payload.

The Entercept Console is very easy to use, providing excellent Agent update, policy deployment and Agent monitoring capabilities for up to 5000 Agents from a single Management Server.

Architecture

With release 4 of the product, Entercept moved from a simple two-tier architecture to a more robust and scalable three-tier system, based on SQL Server as the underlying database. There are now four main components that make up the Entercept system:

- *Management Server*
- *Database*
- *Agents*
- *Console*

Management Server

The Management Server communicates between components of the Entercept system via securely-encrypted SSL sessions. All system activity is logged by the Management Server to the database, and information in the database that is necessary to the administrator is communicated by the Management Server to the Console.

Any system configuration initiated from the Console is communicated by the Management Server to the affected components of the Entercept system.

Communication between components of the Entercept system may be an Agent reporting the trigger of a signature, or the Console reporting a change to a security policy.

Database

The SQL Server database is a repository for all information used by the Entercept system. This information includes a comprehensive list of signatures, security events reported by Agents, system events, user group information, an address book of people who are notified in the event of an attack, and the available versions of Entercept Agent software.

For smaller - or trial - implementations, the cut-down MSDE (Microsoft Database Engine) can be used instead of SQL Server.

Agents

Entercept Agents are installed on every host to be protected in order to provide a layer of protection that identifies and prevents malicious attempts to compromise that host.

The Entercept Agents use a database of security signatures covering a number of different exploit categories, such as buffer overflow, privilege escalations, OS hardening, and so on.

The Agents and the signature database are updated frequently by a “smart update” process that enables automatic update of both Agent code and the signature database.

Entercept Agents also protect specific applications against any operation that is outside the normal behaviour of that application, and protect application resources (such as files and registry keys). Each Entercept Agent protects the Operating System itself and the Entercept Agent and Console applications. In addition, there are Agents that protect specific Web and database servers. The Agent types include:

- ***Windows Standard Edition -*** *protects the OS and the Entercept applications only.*
- ***Windows Web Edition -*** *as Standard Edition, but adds protection for the IIS Web Server.*
- ***Windows Database Edition -*** *as Standard Edition, but adds protection for the MS SQL Server 2000.*
- ***Windows Web and Database Edition -*** *as Standard Edition, but adds protection for the IIS Web Server and MS SQL Server 2000.*
- ***Solaris Standard Edition -*** *protects the OS and the Entercept applications only.*
- ***Solaris Web Edition -*** *as Standard Edition, but adds protection for the Apache, Netscape Enterprise and iPlanet Web Servers.*
- ***HP Standard Edition -*** *protects the OS and the Entercept applications only (new for this release).*

Each Agent has a minimal footprint and communicates with the Management Server via SSL-encrypted sessions. In all cases the Management Server is the *slave*, and the Agent the *master,* meaning that it is the Agent that initiates the connection to the Console.

Thus there are no open listening ports on the Agent hosts (other than for the “nudge” feature mentioned below).

The Agent contacts its designated Console at regular intervals controlled by a heartbeat parameter. This is now set automatically depending on the number of Agents reporting to the Management Server, but this can be overridden manually on larger networks with many Agents, where chatter between Agent and Management Server could result in significant network traffic. In the latest release, there is also the ability to “nudge” the Agent from the Management Server in order to enforce a change of configuration immediately, without having to wait for the heartbeat. This still does not initiate a session with the Agent, but merely prompts the Agent to contact the Management Server sooner that it would otherwise.

If communication with the Console is interrupted for any reason, the Agent will show as “*not connected*” at the Console, but will continue to operate in a stand-alone manner. Security events will be stored locally in an encrypted form until connection is re-established, though if the local storage capacity is exceeded (limited to 10MB by default, but configurable beyond that) events will no longer be recorded. In this case, however, attempted exploits will still be prevented providing the Agent is set to *Protection* mode.

Each Management Server is able to manage up to 5000 Agents, a significant increase over previous releases (no doubt due to the additional tier in the management architecture). If more Agents are needed, additional Management Servers will be required, though there does not appear to be any way to load balance or consolidate reports across multiple Management Servers.

Console

The Console is a Graphical User Interface (GUI) application that allows an administrator to monitor Agent and System activity, manage security, and manage configuration information.

From the Console, it is possible to monitor security events (triggering of signatures, reactions taken by Agents, etc.), system events (detection of new Agents, update of Agent software, etc.), manage Agents (define Agent groups, set the Agent mode of operation, etc.), define and apply security policies, create exceptions to particular events, view and modified severity levels of specific signatures, create custom signatures, manage the versions of Entercept Agent software on the system, manage users, import and export entities, and create reports.

Besides the four major installed components of the system, the actual program that drives the system to perform the functions and protection for the host is comprised of configurable aspects of the system that determine how effectively Entercept protects that host. In addition, the results of this protection are available to view and interpret, to further enhance host protection. These elements are:

- *Events*
- *Exceptions*
- *Policies*
- *Signatures*
- *Notifications*
- *Reports*

Events

Events are viewed on the Console and are the results of actions occurring on the protected hosts. There are two types of events that the system generates: *Security Events* and *System Events*.

A *Security Event* is generated when an Agent recognises a signature or a behavioural rule violation. The event is sent to the Console via the Management Server and logged in the Security Event Monitor (viewed via the Security Event tab).

A *System Event* is generated when a defined set of Entercept system activities occur. System Events are logged in the System Event Monitor and view in the System Event tab.

Exceptions

An *Exception* is a mechanism for overriding a security policy under specific circumstances. In some cases, behaviour that is defined as an attack might in fact be part of a user's work routine or activity that is legal for a protected application.

To allow the user or application to proceed under these circumstances without sending an alert, an Exception can be created. An Exception states, for example, that for a particular Agent, Agent Group, Signature, User, User Group, or Process, the event is ignored.

Policies

Security Policies define the Agent reaction when that particular Agent recognises a signature of a particular severity level. Three possible reactions are taken by an Agent:

- ***Ignore -*** *the event is ignored*
- ***Log -*** *the event is logged*
- ***Prevent -*** *the event is logged, and the specific operation is prevented from taking place*

A Security Policy may state, for example, that when members of a particular Agent group recognise a signature of an *Info* (white) level, they log the occurrence of that signature and allow the process to be handled by the operating system. However, when they recognise a signature of a *High* (red) level, they prevent it from taking place. A Security Policy also defines the type of notification that is generated in response to an event.

Signatures

Signatures are descriptions of security threats and attack methodologies. These may range from installation of unauthorised software to blatant attempts to damage a host. Signatures are categorised by severity level and a description of the danger an attack poses to a host, and each signature is allocated a default Severity Level.

Each administrator can, if required, set different Severity Levels for individual signatures via the Console, and it is also possible to disable a signature so that it is ignored by Agents.

There are four Severity Levels:

- ***Info (white) -*** *Indicates a modification to the system configuration that might create a benign security hole, or an attempt to access sensitive system information. Events at this level occur during normal system activity and generally are not evidence of an attack.*
- ***Low (yellow) -*** *Indicates a modification to the system configuration that might create a more severe security hole than an event of an Info (white) level, or is an attempt to access sensitive system information. Events at this level are not identified as known attacks, but indicate suspicious behaviour on the part of a user or application.*
- ***Medium (orange) -*** *Events at this level are either known attacks with low to medium risk or an indication of highly suspicious behaviour on the part of a user or application.*
- ***High (red) -*** *Events at this level are known attacks that pose a serious threat to a system.*

Signatures can be designed for specific applications - for example, Web Servers such as Apache, IIS, and NES/iPlanet - or for the Entercept Agent or Console. The majority of signatures protect the whole operating system, while some protect specific named applications.

Notifications

The Entercept system allows the administrator to define four types of *Notifications* as part of a Security Policy:

- ***E-mail -*** *E-mails are sent to alert individuals about the event*
- ***Pager -*** *Individuals are paged when an event occurs*
- ***SNMP trap -*** *Traps are sent to a third-part management console when an event is triggered*
- ***Spawned process -*** *A separate process is generated when an event is triggered*

The Address Book is used to create a list of personnel who are to be notified in an event of an attack. Notifications are sent according to specific Severity Levels, Agent groups, or signatures (in previous releases, it was by Severity Level only).

For example, a High level notification may be sent to the Security Administrator via e-mail and pager, whilst High and Medium notifications are sent to other personnel via e-mail only. All SQL Server-related security events can also be e-mailed to the Database Administrator.

Reports

Reports enable the administrator to specify and extract information from the Entercept database.

Reports can provide all the data available for a particular subject (for example, security events) or they can be filtered to deliver specific subsets of that data (*High level events* reported by *X Agent Groups* for a specified *time period*, for instance).

How Does It Work?

Entercept is a kernel-level security technology, although it does not actually modify the OS kernel in any way. Instead, the Agent installs itself just above the kernel, where it can intercept System and API calls, understand their parameters and context, evaluate them in real-time against malicious behaviour, and then allow them to be processed by the OS, or reject them.

As with normal applications, all exploits need to use OS resources in order to achieve anything. Therefore, the System Call/API Call interface is the logical place to reside in order to have a complete view and understanding of a machine's processing environment. In order to be proactive, a protection system needs to reside at this level if it is to have the capability to prevent hacks in real-time.

The context of System and API calls is well-defined – they are used in a well-documented way, with well-defined parameters, in order to achieve specific and identifiable results. Because of this, there is little ambiguity in determining whether a request for system resources can be termed "*good behaviour*", or whether somebody is trying to exploit a known vulnerability. In addition, nothing is encrypted at this level, meaning that all parameters are passed in clear-text, making it easier to identify malicious intent and keeping false positives to a minimum.

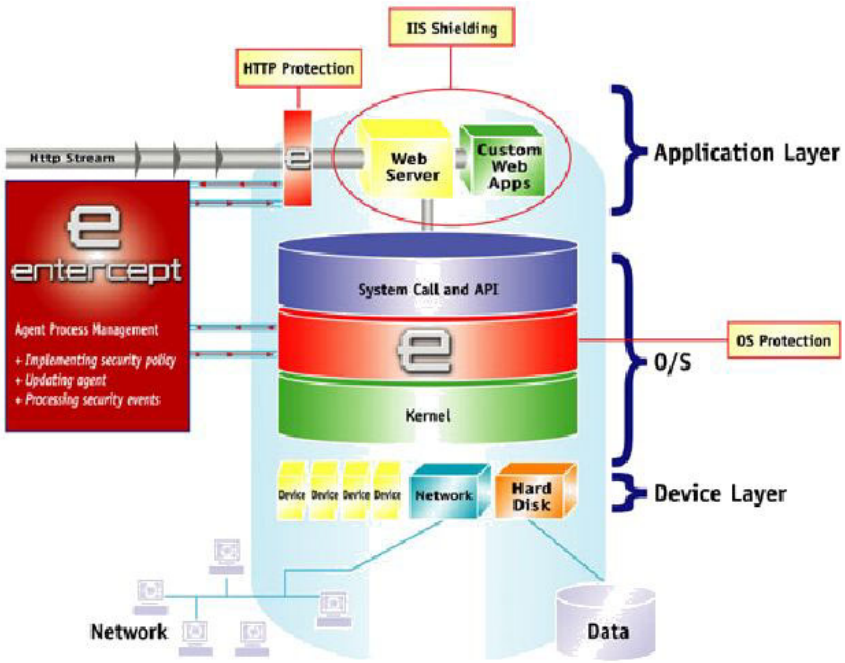

Figure 1 - Entercept: Architecture

Entercept is available in three versions: the *Standard Edition*, the *Web Server Edition*, and the *Database Edition*. The Web Server Edition includes all the functionality of the Standard Edition as well as additional features specific to preventing attacks against Web Servers.

Entercept Standard Edition

The *Entercept Standard Edition* protects the most important part of any server: the operating system.

All users and programs access the server through the operating system. Entercept Standard Edition runs on Solaris, HP-UX, and Windows, and offers the following capabilities:

- **Resource Protection -** The Standard Edition protects system resources (libraries, files, directories, user accounts) and prevents them from being amended in any way, thus preventing Trojan horses, rootkits, and backdoors from altering the system resources in order to install themselves.
- **Prevention Of Privilege Escalation Exploits -** Privilege escalation attacks are designed to give ordinary users super user-level (root or administrator) access to the server. The Entercept Standard Edition prevents these attacks from succeeding by preventing access to the files and resources necessary to alter privilege levels. Even new, previously unpublished privilege escalations can be stopped without knowledge of the specific exploit. This is possible since all privilege escalation exploits alter user privileges, and Entercept prevents such alterations.
- **Buffer Overflow Exploit Prevention -** The Entercept Standard Edition is able to determine if code that is about to be executed by the OS came from a normal application or from an overflowed buffer. If the code came from a normal application, Entercept allows it to be executed. If it came from an overflowed buffer, it is blocked, and the buffer overflow exploit is thwarted.
- **Unknown Attacks -** Entercept can prevent the aforementioned attacks using behavioural rules technology, rather than relying solely on individual signatures. This technology allows Entercept to stop new and previously unknown attacks without requiring signature updates to the product. For example, Entercept's rules to stop buffer overflow exploits from succeeding are not tied to a specific application or signature. Instead, Entercept can prevent buffer overflow exploits from succeeding, regardless of the application or buffer involved.
- **SecureSelect -** Entercept provides three security modes: *SecureSelect-Warning Mode, SecureSelect-Protect Mode,* and *SecureSelect-Vault Mode*. Each mode provides more security than the previous one. Customers begin Entercept deployments in Warning Mode, then progress to Protection Mode, and Vault Mode as they tune and tighten their Entercept installation.

Entercept Web Server Edition

The *Entercept Web Server Edition* (WSE) runs on Solaris and Windows. It supports *Netscape Enterprise Server, iPlanet Web Server, Apache Web Server,* and *Microsoft IIS Web Server*. The Web Server Edition includes all the functionality of the Standard Edition as well as additional levels of protection specifically tailored for Web servers, including *HTTP Filtering* and *Web Server Shielding*:

- **HTTP Filtering -** Entercept Web Server Edition includes an HTTP filtering layer that intercepts HTTP requests after they are decrypted and decoded (from any SSL encryption, Unicode encoding, or hex encoding) before the Web server executes them. Entercept uses signatures at this layer to detect attacks against the Web server and other vulnerabilities.

This layer is the preferred place to stop attacks, since they are blocked long before the server can actually execute them.

- **Web Server Shielding -** Entercept also employs Web Server Shielding to stop both known and unknown attacks from altering Web content or using the Web server as an attack tool.

  Entercept places the Web server application, its files, and its resources inside a virtual “steel vault”. If the Web server attempts to access any resources outside that vault, Entercept blocks the attempt. Conversely, if any other user or process tries to access or alter the files or resources contained within the vault, Entercept blocks that access as well.

  Entercept accomplishes this protection by defining a set of behavioural rules for the Web server. If the Web server attempts to do things that are not within its defined behaviour, the attempt is blocked. This ensures the integrity of the Web server, its applications and files (including customer data), and enables Entercept to protect Web servers from both known and unknown attacks.

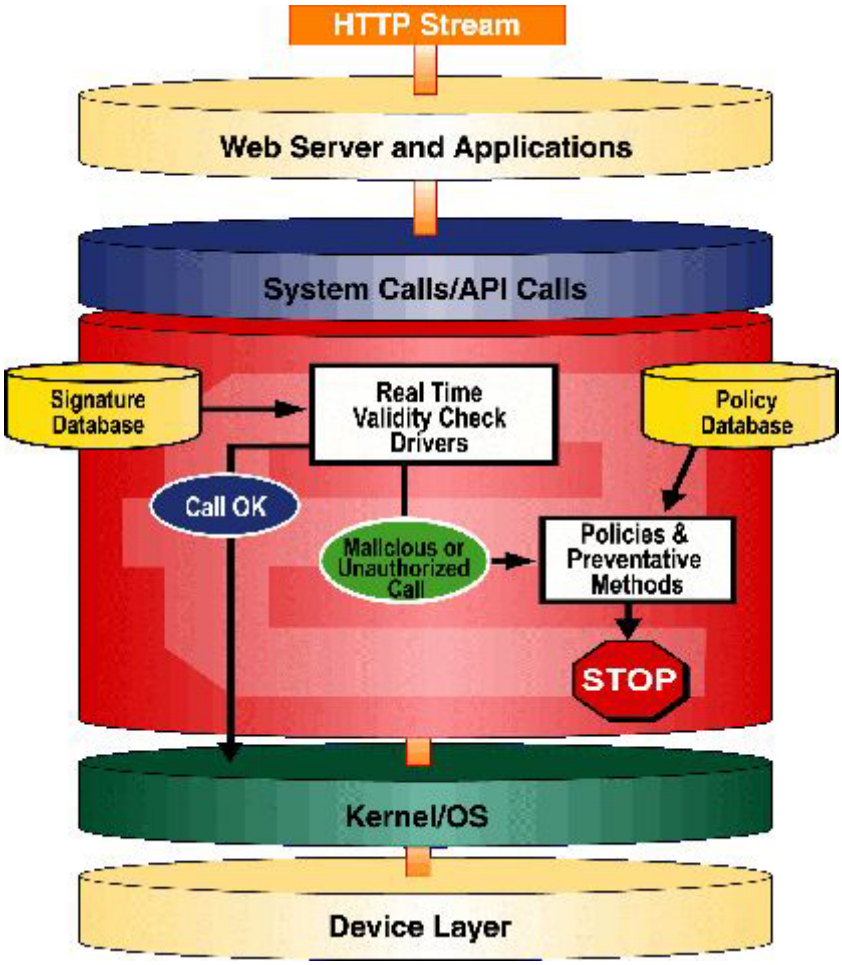

Figure 2 - Entercept: Web Shielding

The shielding concept is based on a simple yet powerful assumption that the Web server and Web application execution pattern is very specific and repeats itself. Thus, it is possible to accurately characterise the “normal” behaviour and the typical access patterns to resources of the Web server.

The “normal behaviour” of the Web server is encoded as rule templates rather than fixed rules, allowing the Web server to be installed and configured in different ways for different installations. As the Agent starts up, the shielding mechanism initiates a scanning process that gathers a set of attributes reflecting the details of the installation and configuration of the Web server, including services, program files, data files, Web pages and Registry settings.

Next, an instantiation program is executed that derives the appropriate rules from the given rule template. The instantiation program uses the information gathered during the scanning phase to fill the missing information within the rule template and to generate the specific optimised rules.

Once the set of rules is ready, they are loaded into the Agent. This procedure is executed whenever the Web server configuration is modified. Entercept's protection tightens the security of the Web server resources to the extent that even if an intruder gains administrative privileges, he still cannot access those resources.

It accomplishes this in a number of ways:

Program File Shielding: The shielding protects the Web server executables and configuration files from being modified or deleted. As an example, the “inetinfo.exe” file, which is the main IIS executable, will be located and an appropriate rule including its current directory will be generated. This rule will not allow any program other than the Windows Installer to modify this executable.

Data File Shielding: Access to the Web server data files is restricted to the process running the “inetinfo.exe” executable. Since the executable is shielded, it is guaranteed that no other process will access the data files. In addition, the static Web pages are also shielded, and any attempt to either modify or delete these pages is prohibited. The only program that can modify these files is the predefined Web-authoring tool run only by the predetermined Web master.

Registry Shielding: To function properly, the Web server relies on settings stored in the system Registry. If the intruder modifies the appropriate Registry entries, he can affect the operation of the Web server. The Registry shielding makes sure that the correct level of read/write access is granted only to the appropriate processes.

Service Shielding: To prevent denial-of-service attacks, the Web server shield includes rules that prevent any attempt to stop the Web server service or change its start-up mode.

User Shielding: The shield makes sure that the privileges of the users under which the Web server runs cannot be modified. This eliminates the possibility of escalating the privileges of the Web server user, a common goal of intruders. The shield also prevents changing the user under which the Web server is running. If, for example, the user was changed to Administrator, ensuing attacks could be harmful since the attacker could execute code with Administrator rights.

Hackers often exploit Web server and application vulnerabilities in order to escalate privilege, execute malicious commands or access data. Since the normal behaviour of the Web server is well defined, most of these deviations can be identified and prevented.

Web shielding ensures that nobody (process or person) can alter the configuration, layout, and operation of the Web site.

Even if intruders gained administrative privileges on the server box, they still would be unable to deface or otherwise modify the content files (such as the corporate Home page).

Entercept Database Edition

Entercept Database Edition provides a means to protect assets and ensure database server integrity by protecting against both unknown and known attacks, including popular SQL Injection attacks. Entercept Database Edition locks down the database to both enforce correct behaviour and block abnormal behaviour

Entercept uses a technique called SQL Interception to intercept all incoming database queries and block any that would result in malicious activity. Entercept's *SQL Interception Engine* analyses the query for buffer overflow conditions, SQL injection attempts, and abnormal manipulation of the database. Using this information, calls are matched against the appropriate behavioural rules and known attack signatures. Entercept then blocks queries that attempt malicious behaviour or match any specific attack signature. All preventive activity is logged to the Entercept Management System for review and reporting.

The Database Edition performs the following operations:

- **SQL Injection Protection** - protects against a common threat to database security: SQL injection techniques. By entering cleverly-crafted SQL statements into a vulnerable application's data fields, attackers can access restricted data such as credit card numbers, delete private data, alter data, and even attack the other computers on the database server's network. The Entercept Database Edition prevents SQL injection attacks by validating SQL queries before they are processed by the database engine. Malicious SQL injection attempts are rejected and the database's integrity is preserved.
- **Specific Attack Prevention** - prevents attackers from disrupting the database. Dozens of known attacks exist that are designed to crash and/or compromise database servers. Using SQL Interception technology, Entercept blocks these attacks before they can cause any harm to the database.
- **Database Shielding** - protects databases and data from unauthorised access. Database shielding ensures that no process, other than the database itself, will be able to access the database's execution environment, data, or settings. In addition, the database is prevented from accessing non-database resources. This prevents attackers from using the database to launch attacks against other targets.
- **All Features of Entercept Standard Edition** - The Entercept Database Server Edition includes the features described above, as well as all the features present in the Entercept Standard Edition: known and unknown attack prevention, buffer overflow exploit prevention, resource protection and prevention of privilege escalation.

One of the key advantages of Entercept is the fact that all activity on the host is seen, and is not impaired by encryption, switched data or reliance on system log information. Entercept functionality can be used to protect specific resources of the OS or applications (registry keys, accounts, files, processes, and so on), and from that point of view, it can be viewed as a means to harden both the OS and applications on a server.

In fact, this “OS hardening” capability has been extended and automated in recent releases of the software through the introduction of the *SecureSelect Vault Mode*. This mode of operation provides users the capability to lock down their operating system based on *Vault Signature* rules, which are designed to monitor for critical activities that would normally be signs of OS update or patching activity.

Performance

The aim of this section is to determine the impact the Host IPS agent has on the host on which it is installed.

Entercept stopped all of the attempted unauthorised access to critical files, directories, registry keys and applications, and it was very straightforward to create exceptions to allow legitimate operations to proceed where they have been prevented in error.

As you would expect from an agent incorporating an ISAPI filter, there is some impact on the overall performance of the host Web server once the Agent software has been installed.

The most obvious impact of the Entercept Agent is to reduce the maximum capacity of the server when under extreme loads, as well as to increase the average page and URL response times. However, when the server with and without Agent is compared under the real-world load levels of our tests, the differences - although **numerically** significant in that the average response time is approximately four times greater with the Agent installed than without - are still only in the order of **microseconds**.

Given that all response times, with or without the Agent installed, remain **less than one millisecond** throughout our tests, it is extremely unlikely that the Agent will make a noticeable difference to the user experience.

Finally, the FTP tests show almost no performance degradation between the two tests. This indicates that the actual impact of the Agent alone is indeed as low as that claimed by Entercept, and that the most significant impact (numerically, rather than real-world) is imposed by the use of the ISAPI filter in Web traffic.

Overall, the Entercept software is unlikely to impact negatively on the average user experience, especially when weighed against the benefits of having the Agent installed.

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

Security Effectiveness

Entercept did well in all our tests, providing the ability to detect most of our activities via the default policies and signatures.

Where our requirements fell outside the scope of the built-in rules, the custom signature capability - which is much more flexible in the current release than in previous versions of the software - allowed us to cover most activities (the only exceptions being custom signature creation for FTP exploits and user-specified network services/ports).

None of our evasion techniques had any effect on the detection or prevention capabilities of the Entercept Agent.

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

Although this is a slightly more complicated than previous releases due to the addition of the Management Server, installation is still very straightforward. All that is required is to run the Installer from the CD supplied and select the appropriate menu options.

The first job is to install the Management Server and its underlying database - this can be SQL Server, which is recommended for all live deployments, or MSDE, which is suitable for trial (or very small) implementations. During installation a public/private key pair and accompanying digital certificate is generated to enable authentication and encryption between Agents and the Management Server.

The Management Server also includes the Tomcat Web server in order to provide secure Web-based access to the Entercept Console. It is recommended that a Web and Database Agent be installed on the Management Server to provide complete protection, and one is included in the package at zero cost.

The Java-based Entercept Console can be installed on any host on the network with access to the Management Server. Once Management Server and console have been installed, it is time to deploy the Agents. The Agent can be installed from the CD or from a network share, and the Management Server's public key must be provided to the Agent during installation. This can also be acquired from the Management Server (in unencrypted form), though the most secure means of distributing this vital component would be via an out of band method such as floppy disk.

Although the initial installation is performed manually, subsequent upgrades to the Agent are carried out from the Console. Organisations with large numbers of Agents to roll out may want to consider the use of third party software distribution tools for the initial installation.

Despite the low-level operation of the Agent software, no reboot is necessary after installation (or uninstallation), and the Agent immediately contacts the Console to determine its initial mode of operation. Parameters in the Console specify whether the Agent should be active by default following installation, and whether it should start up in "*Warning*" or "*Protection*" mode.

All new Agents have the default policy applied automatically, and each Agent is brought up in "*Warning*" mode to ensure that alerts are raised immediately, though no suspicious operations are blocked.

Configuration

The Console is protected by a user name and password combination, and any number of users can be created. The options and views available to a user when using Entercept are based on the *role* they are assigned by the system administrator. Each role (except for the *Global Administrator*) only allows certain actions to be performed, eliminating Console menu tabs and restricting the scope of alerts that can be viewed where necessary.

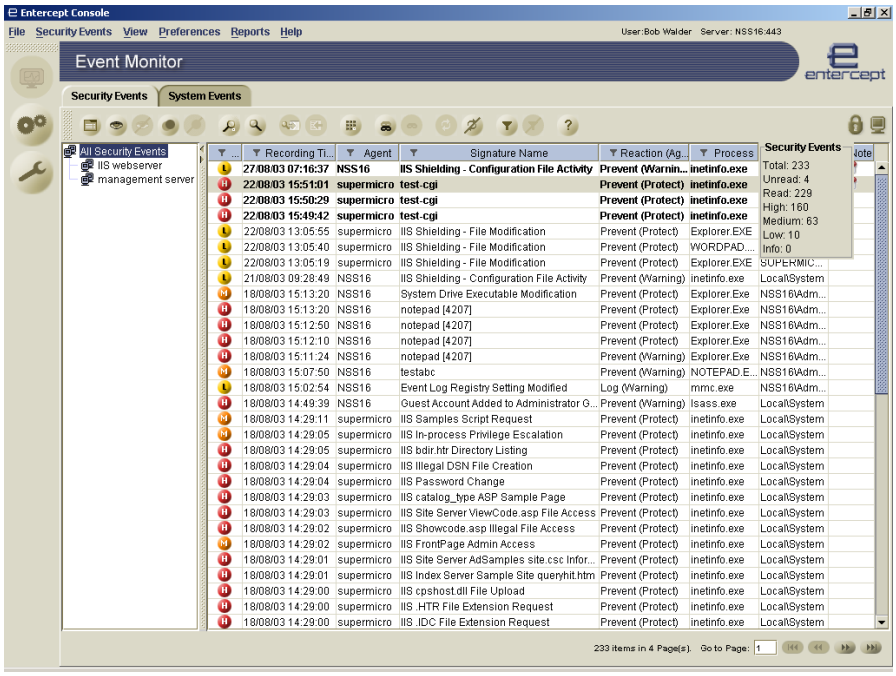

Figure 3 - Entercept: Java-based GUI Console

In addition, each user can be restricted to working on one or more Agent groups if required. For example, a user with a *Viewer* role can create a report for the Agent group they are associated with, or can view security events, but they can not create Exceptions or Policies. This prevents unauthorised users from editing entities that can adversely affect the way the system operates. Roles allow the system administrator to enable only certain qualified users to manage sensitive Entercept features.

All critical configuration files and databases are protected by the Entercept Agent itself, a copy of which can be installed on the host machine following installation of the Console software (a free license is provided for this Agent).

The Console has changed somewhat in appearance - though not much in the way it operates - since the last time we looked at this product in our labs. This is because the GUI has been completely redeveloped as a Java application. The main Console screen is clear and easy to read, with a toolbar down the left hand side consisting of three icons: *Event Monitor, Configuration* and *Tools*. On selecting one of these options, a tabbed window appears in the right hand pane, once tab for each of the options in that menu.

One nice ease-of-use feature is that the right hand pane is itself divided into two sections.

The main operations always occur in the right-hand section, whilst the left-hand section always contains a hierarchical tree view of the elements which are being operated upon. So, for example, if the administrator is working on Signatures, the tree will present a top-level view pertaining to “All Signatures”, and lower-level branches which correspond to Agent Groups, thus allowing the administrator to quickly select Signatures applied to an individual group only. The *Address Book* tab has a top level of “All Contacts”, and lower-level branches for each type of contact, such as e-mail, pager, SNMP trap, and so on.

Every window in the system which displays data allows that data to be sorted by any column, and columns can be moved around (though not deleted) to format the screen layout as required. An excellent new feature is the provision of a *filter* icon on every column of data in the system. By clicking on this icon in a particular column, the administrator is presented with a drop-down menu of selections (if the column contains a range of fixed values, such as *Severity Level*), or a free-format text search box.

By entering values within these search boxes, the data is filtered and the only records displayed are those which match that particular criterion. The icon changes colour to indicate that a filter is active on that column, and multiple filters can be active on multiple columns.

Alternatively, selecting any row of data and then clicking on the filter icon will set the filter immediately to the appropriate value in the row selected. A separate filter dialogue window is also available to allow multiple filters to be set in a single place if required.

This feature is extremely easy to use, and yet extremely powerful in operation, enabling the administrator to quickly and easily isolate small subsets from large volumes of data, whether it be configuration data such as signature definitions, or alert details.

Policy Management

The *Policy* option is something of a misnomer, since unlike a typical IDS or IPS product, Entercept does not expect the administrator to define which signatures apply to which Agents. Instead, **all** enabled signatures are applied to **all** Agents by default (some of the more “noisy” signatures are now disabled by default out of the box), and the Policy simply determines how specific Agents or groups of Agents will react to security events, and which people will be notified of those reactions. The definition of how an Agent will react is based on the *Severity Level* of each security event.

All that is required is to define the Agents and/or users to which the Policy applies. Policies can be as granular as they need to be - applying to a single host or user if required - since it is possible to apply multiple Policies to individual entities. Where conflicts occur (where a group-based Policy has different settings to an individual one, perhaps) they are resolved automatically by Entercept applying the highest security setting of those that are in conflict.

Each alert generated by an Entercept Agent has a *Severity Level* associated with it - *high*, *medium*, *low*, *info* or *disabled* - and this determines the action that is taken by the Agent when each alert is raised.

Three direct actions are supported:

- **Ignore -** the action that caused the alert is allowed to continue and no record is kept
- **Log -** the action that caused the alert is allowed to continue, but details of the alert are recorded
- **Prevent -** a log entry is made and the action is terminated. A sensible error code is returned to the application that caused the alert so that it is not obvious that it is an IPS that has terminated the action.

In addition to one of the direct actions above, it is also possible to generate alerts via e-mail, pager, spawn process (which could also take additional corrective action via external programs, scripts or batch files), or SNMP trap. Log entries are always recorded to the Entercept database and reported directly to the Console screen.

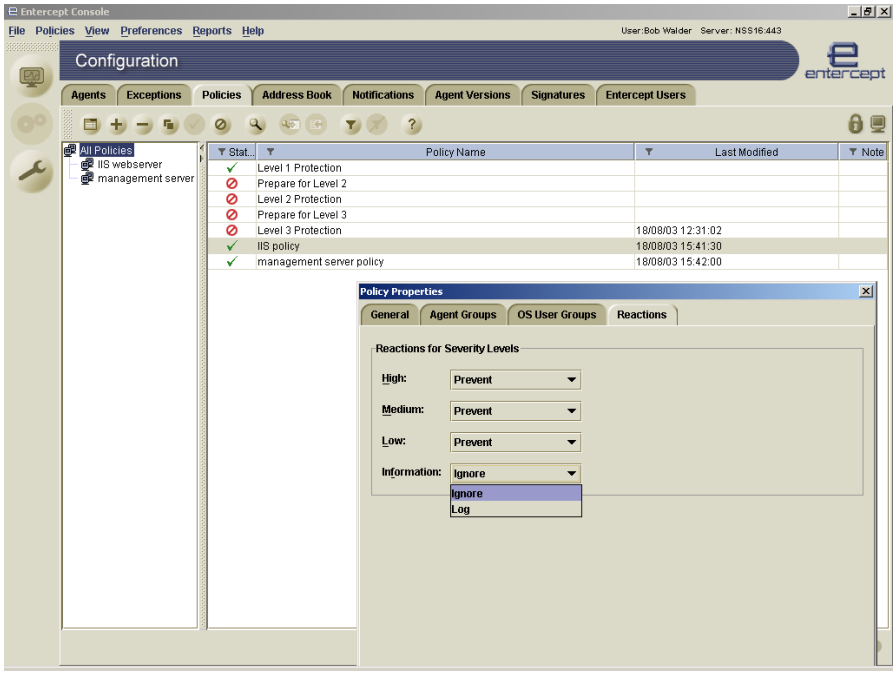

Figure 4 - Entercept: Policy management

In the latest release, the product includes a small number of pre-defined Policies which are created during installation. These range from *Level 1* - where only *High* level security events are prevented (this replaces the old "default" Policy in the latest release) - to *Level 3* - where all levels except *Information* are prevented - and are designed to allow administrators to be up and running as quickly as possible. In addition to the noisier signatures being disabled by default, many of the other signatures have also been re-graded for this latest release in an attempt to ensure that High (red) signatures are now only comprised of events with zero chance of raising a false positive. Thus, the Level 1 policy can be selected with confidence from day one.

However, should the administrator wish to "re-grade" a number of signatures himself (perhaps to re-enable all of those which are disabled), there is a problem. It is not possible to make bulk changes to Signatures in this way - each one has to be amended separately. This makes operations such as changing all "*Disabled*" signatures to "*Informational*", a long winded affair. In addition, the Signature tab still only shows the default Severity Level, and not the new one.

This is partly necessary because it is possible to apply multiple Severity Levels per signature - one for each Agent group, for example - but it still makes it very difficult to track changes at a glance.

The *Agents* tab provides a list of Agent groups down the left, and Agent details on the right. It is the Agent Groups that determine the security policy that is applied and, as mentioned previously, it is possible to allocate Agents to more than one group. All new Agents appear in the "*All Agents*" group by default, from where they can be moved into other groups as required.

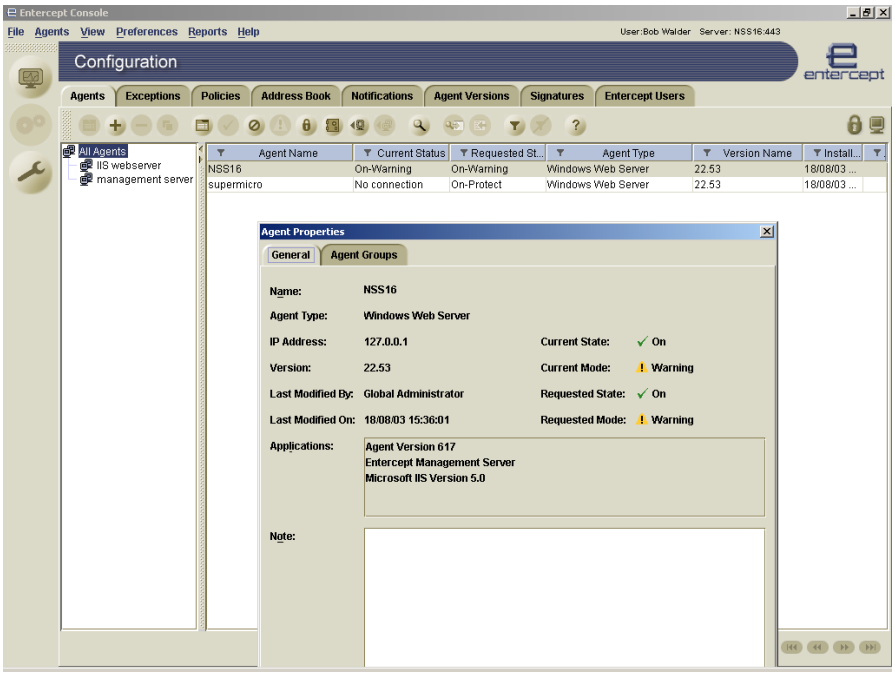

Figure 5 - Entercept: Configuring Agent properties

Selecting any Agent Group brings up a list of Agents within that Group, with details of the Agent type (OS platform, and whether it is a Server or Web Server Agent), version, current state and requested state. The Console can check the Entercept Web site at regular, administrator-defined intervals and automatically download new versions of the Agent software.

With the latest release, it is possible to have multiple versions of an Agent active on the network at the same time (previous releases allowed only two versions to be active - a "*live*" and a "*test*" version). This provides the means for the administrator to test new Agent releases on specific test machines before deploying them live across the organisation.

Once correct operation has been verified, the new versions can be rolled out to all the hosts controlled by the Console automatically - no reboot or other intervention is required on any of the Agent hosts. It is also possible to roll-back versions to a previous release at the click of a button on the Console should problems arise.

The "state" of an Agent can be set to one of three values, which are known as *SecureSelect* levels:

- **Warning Mode -** Entercept logs all perceived malicious activity, but does not block that activity.

This mode is very useful for tuning the Entercept system and assessing Entercept's impact on installed applications. that would otherwise violate the Entercept Security Policy. Note that even in Warning Mode, there are some events - such as trying to delete the Entercept database - which will always be prevented.

- **Protection Mode -** This is the next step in the deployment lifecycle. Once any tuning has been performed in *Warning Mode*, security administrators can enable *Protection Mode*. This mode protects against buffer overflow attacks, specific known attacks, and enforces behavioural rules on the system to prevent new, previously unknown attacks.

  Those events which are marked as *ignore* or *log* are allowed to pass, whereas those marked as *prevent* will be stopped and logged immediately by the Agent. *Protection Mode* provides the full protection of Entercept, without the extra OS lock-down features provided by *Vault Mode*, and thus requires less tuning. Organisations may elect to remain in *Protection Mode* or continue to improve their security posture by moving to *Vault Mode* as required.

- **Vault Mode -** This mode is designed to build upon the protection present in *Protect Mode* and take it one step further. *Vault Mode* locks access to the key files and settings most critical to the operating system and prevents them from being accessed or changed, even by users with root or administrator privileges. This prevents attackers from compromising the OS, even if they are able to obtain root-level access.

  Vault Mode is most appropriate for servers that do not change frequently, as Entercept will prevent all attempts to change the system's configuration - including applying OS updates and patches. Necessary, authorised configuration changes can be accommodated via Entercept's exception mechanism or by placing an Agent temporarily in *Warning Mode*.

By locking down the OS critical files, Entercept is attempting to provide some of the advantages of a *trusted OS* without the management and cost implications associated with replacing existing operating systems. Attackers are unable to install rootkits, Trojaned versions of system files, or viruses on Entercept-protected servers that are Vault-enabled. Additionally, by protecting and locking the critical OS configuration files, Entercept thwarts attempts at server compromise by changing the OS configuration.

Naturally, such a complete lock-down of the OS can also have tremendous negative impact too, so the administrator needs to be sure that Vault-enabled servers are not running - or accessed by - software that may need to access or update critical OS files or settings for any legitimate reason. Vault Mode would be particularly useful for dedicated Web and FTP servers, for example, particularly those installed as public-facing servers in the De-Militarised Zone (DMZ).

The pre-defined policies are always applied automatically from the moment the Agent is activated although, by default, Agents are initially placed into *Warning Mode* only (this default initial setting is configurable). However, now that *High* (Red) signatures are considered safe to deploy immediately with zero chance of false positives, the administrator is able to move to a basic Protect Mode level of operation (at least with the default *Level 1* Policy) very quickly.

Administrators are advised to run Agents in *Warning Mode* for a short period of time following installation to provide some idea of what could be considered “normal” behaviour. The information that is gathered while the system is in Warning Mode can then be used to create any *Exceptions* that may be needed before moving to *Protect Mode*. Entercept Exceptions enable customisation of the predefined Entercept Security Policy and allow custom applications to perform any necessary functions that would otherwise violate the Policy.

For example, Entercept can log details of any attempt to add, delete or modify files in the protected Web directories. However, these are all normal activities for the Webmaster, and so it would be desirable to define *Exceptions* to these events. An *Exception* allows the administrator to specify when a particular security event can be ignored in order to filter out false positives, and it can be applied on an individual Agent, user or process basis if required. This would allow a rule which says that files can be added to the Web root directory only if they are added by user *Webmaster*, using *FrontPage* on server *WebServer*.

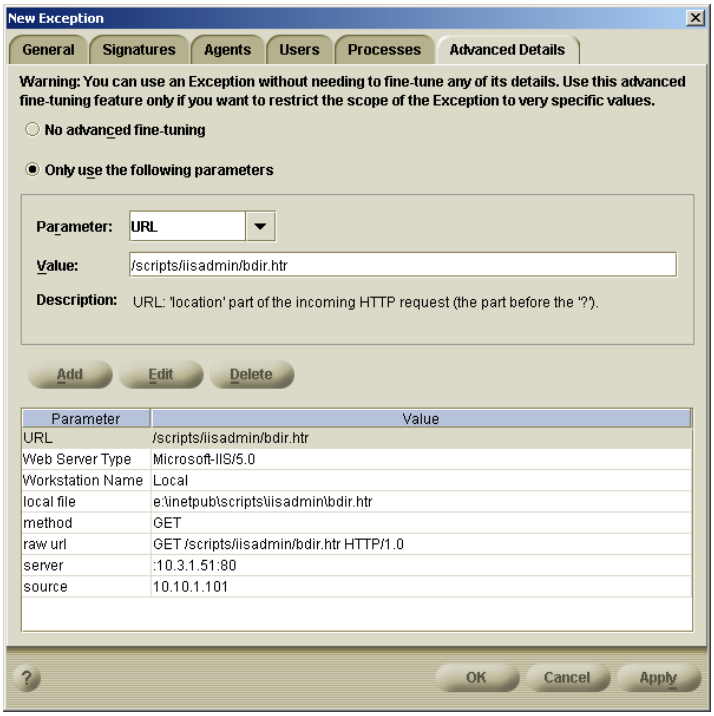

Figure 6 - Entercept: Advanced Exception properties provide increased flexibility

Exceptions can be defined based on a number of parameters, including users and groups, processes (applications), Agents, specific Signatures, and “Advanced Details” (wildcards are acceptable). That last one is extremely powerful, since it allows Exceptions to be based on one or more pieces of “context data” from an event, such as the source or destination IP address, workstation name, URL and so on. It is also possible to combine multiple signatures into a single exception.

Without a doubt, the easiest way to create an Exception is to wait for a security event to be raised, right click on the event and select “*Create Exception*”, and then tune the Exception parameters as required. When reviewing security events raised from an Agent in Warning Mode, this provides a relatively quick and easy means for the administrator to fine tune a Policy before switching the Agent to Protect Mode.

However, it does have the disadvantage of actually requiring a security event to be raised in the first place, and this may not always be desirable (especially when the Agent is already in Protect Mode and the administrator knows an Exception is required to prevent a new application from being blocked). For these cases, it is possible to create Exceptions from scratch in advance, so that where the administrator knows that a Policy rule will cause problems, those problems can be pre-empted and the Policy tuned *before* it is deployed.

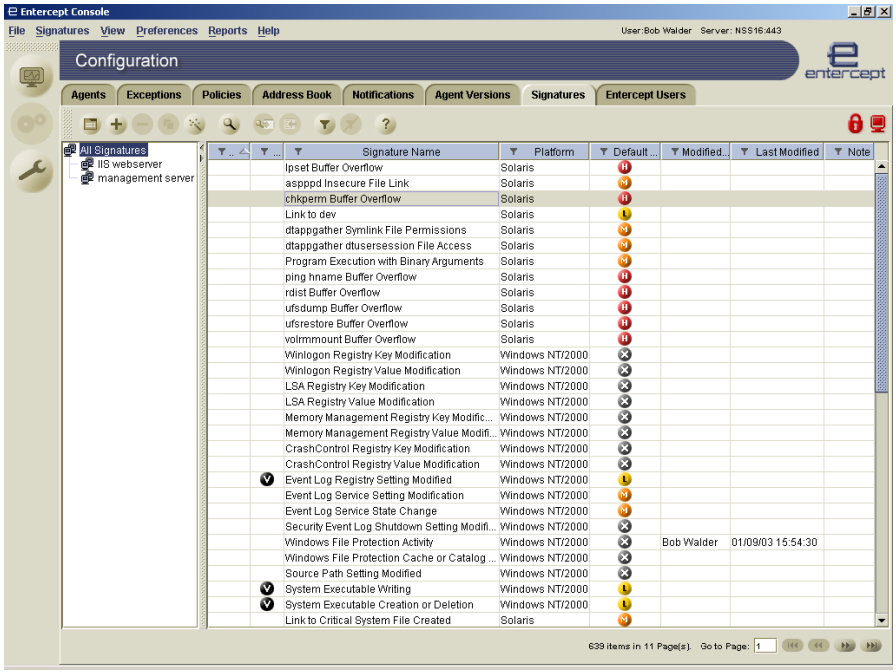

Figure 7 - Entercept: Signatures

The heart of the Entercept system consists of the *Signatures*. These are descriptions of security threats and attack methodologies, ranging from installation of unauthorised software, to attempts to damage a computer. Signatures consist of a set of rules, or entities, associated with a *Severity Level* that defines a set of possible occurrences or incident. Severity Levels are divided into four colour-coded categories (*red, orange, yellow* or *white*) indicating the different levels of potential danger to a system (*high, medium, low* or *info*), and allowing the administrator to define different reactions to different levels of potential harm. When a Signature is activated, the Policy is checked to determine what action is to be taken, as defined by the Security Level. A Signature can also be disabled, so that it is ignored by Agents. Signatures can also be designed for specific applications, for example: Web Servers such as Apache or IIS; Database Servers such as Microsoft SQL Server 2000; or the Entercept Agent or Console.

Severity Levels can be modified for individual Signatures (there is no means to alter Levels for a group of Signatures) and thus define how Entercept will respond to specific security alerts. For each Signature, it is possible to override the default Security Level on a global basis (i.e. for all Agent groups) or for a specific Agent group.

For example, if an attempted directory traversal is detected on an Internet-facing Web server, it could be designated as "*High*", whereas on a purely internal server it would be "*Medium*".

Whereas "*Medium*" priority would ensure that the event is prevented and logged, "*High*" priority may also reprogram the firewall (via a spawned process) to prevent further traffic from the attacking host, as well as raising an SNMP trap to escalate the alert to a central network management console.

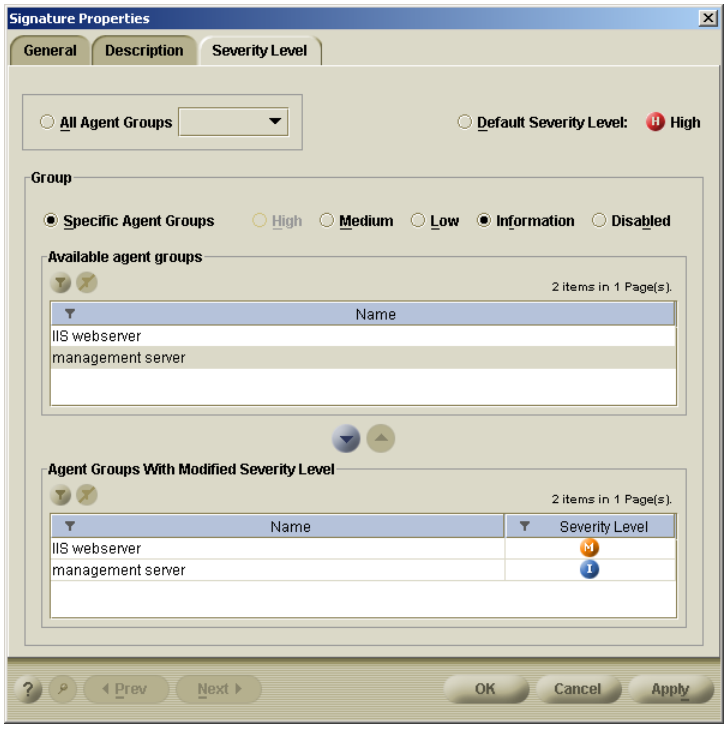

Figure 8 - Entercept: Signature Severity Levels

Security Levels provide an enormous amount of flexibility in how attacks are handled within Entercept, but once a large number of them have been customised it can be difficult to determine exactly what constitutes the actual Security Policy applied to any given Agent or Agent group.

One of the biggest changes in the recent releases has been the ability to define custom Signatures from scratch. This has been further enhanced in the latest release via the addition of GUI tools to define and maintain those custom Signatures (previously handled via a cumbersome method of editing text files). This allows Entercept to compete more readily in the traditional Host IDS market place since it is now possible to monitor and prevent access to any file or registry setting on the protected host.

The rules definition capability is very flexible - each rule can be designated as an *include* or *exclude* rule, and wildcards can be used throughout. This makes it possible for Entercept to monitor and/or prevent access to a wide range of files, directories, applications and registry entries. Signatures can be created from scratch for those conversant with the rule definition language, or via a graphical Wizard. The latter is the simplest way to create a basic rule which can then be tweaked and tuned manually as required.

Once custom Signatures have been created, they are treated like normal rules in terms of policy definition and alert generation, although they are displayed with a "*Custom*" icon that distinguishes them from standard signatures in the GUI. As with normal Signatures, it is possible to define Security Levels and Exceptions for custom Signatures.

When using more than one Console, the *Import and Export* feature is a new feature that can be used to transfer the following information between them:

- *Agent Groups*
- *Policies*
- *Address Book*
- *Notification*
- *Exceptions*
- *Signatures*

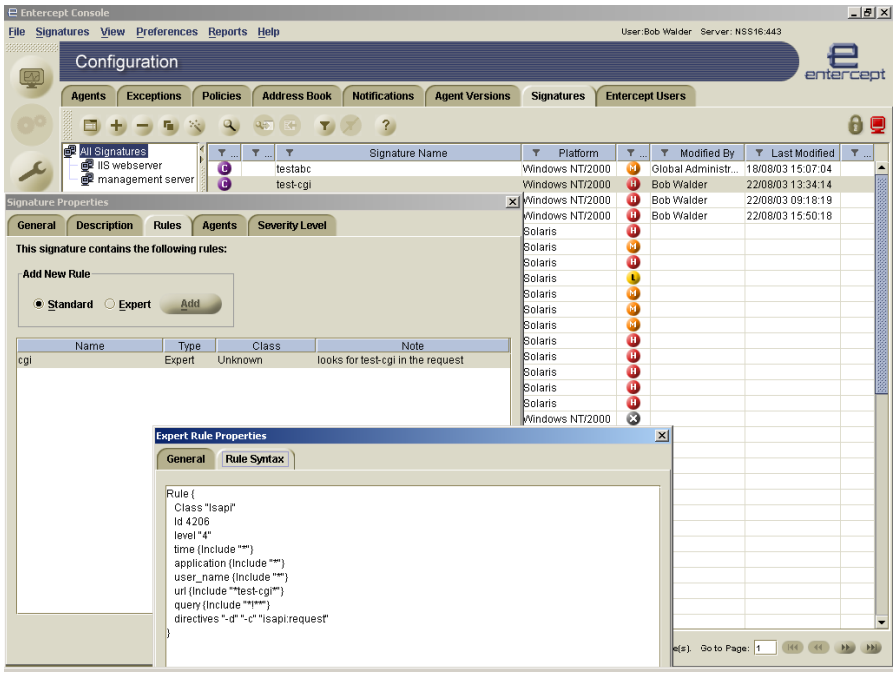

Figure 9 - Entercept: Creating Custom Signatures

Transferring information is performed through an *export basket* that contains the objects that are being imported and exported in a configuration file. These objects contain related entities. This feature allows the administrator to export all local configuration (useful for backup and Console duplication), certain information (all exceptions, security policies, and so on), or specific settings.

Alert Handling

The Console provides the *Event Monitor* window, which contains two real-time monitoring tabs (one for *Security Events* and one for *System Events*) to display events as they are transmitted from the Agents.

The *Security Event* tab lists all Security Events reported by the various agents that report to the Console. It provides a one line description of each event, and - as with other areas of the system - all the events can be filtered or sorted on any of data in any of the columns. There is also a comprehensive search capability allowing subsets of the available data to be displayed based on various criteria such as *Severity Level, Signatures, Agents, Processes* and *Users*. The filtering, sorting and searching capabilities - some of which are new for this release - make tracking down similar or related groups of events very straightforward.

By clicking at any point in the hierarchical tree view to the left of the screen it is also possible to view Events from all Agents, or only those Events raised by a specific Agent.

Events can be marked as read or unread, and it is possible to store notes against each event. Events can also be hidden once they have been dealt with in order to reduce on-screen clutter, and unhidden should further investigation be required. One bad point about the interface is that when using the “Select All” option to mark as *read* or *hidden*, only the items on the current page are selected, and not the entire set of unread/unhidden alerts. This can make it a long and laborious process to clear a large number of alerts.

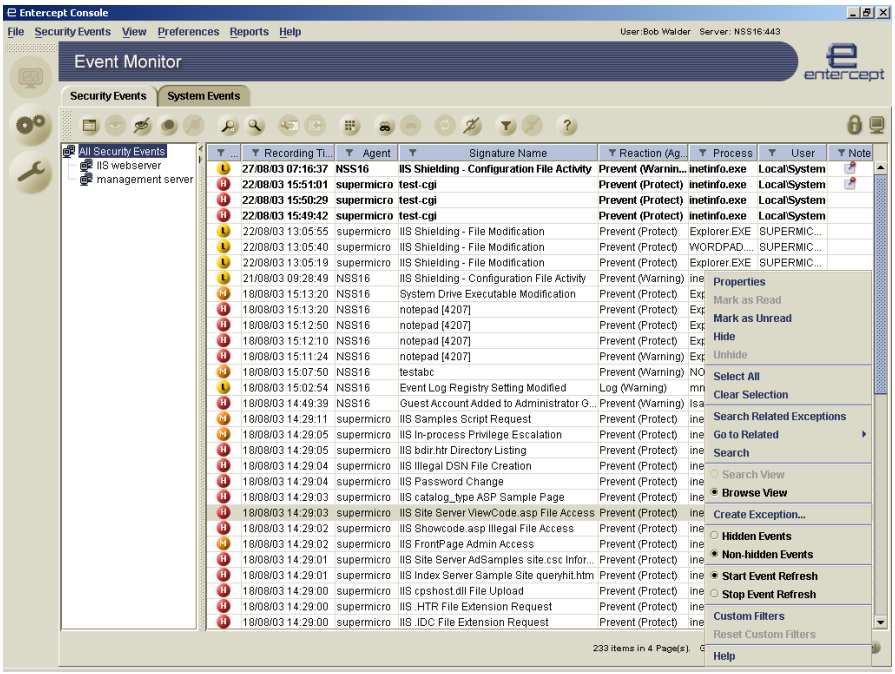

Figure 10 - Entercept: Handling alerts via the Security Event Monitor

Note that when two events are triggered by the same cause, the reaction taken is the highest of the two. For example, assume that *Low* level events are assigned the reaction *Log*, and *Medium* level events are assigned the reaction *Prevent*. When these events are triggered by the same cause, the reaction is *Prevent*.

Two small icons at the top right of the screen provide an instant indication of the arrival of new Security or System events. On clicking these icons, the status is cleared and a window popped up to provide a quick count of the total number of events of each Severity Level.

More detailed information can be displayed for each event by viewing its properties, where the event and the actions that triggered it are described fully. If it is decided that a particular event is a false positive, a click of a button is all that is required to create an exception based on that event.

The *System Event* tab is very similar in appearance and operation to the Security Event tab. It’s job, however, is to monitor Entercept’s system activity, such as services starting/stopping, console administrators logging on/off, agents starting/stopping, and so on. Events can also be filtered in a similar manner to the Security Event tab.

Reporting and Analysis

Reporting continues to improve with each release of the software, and there are a number of pre-defined text and graphical reports available via the *Reports* menu.

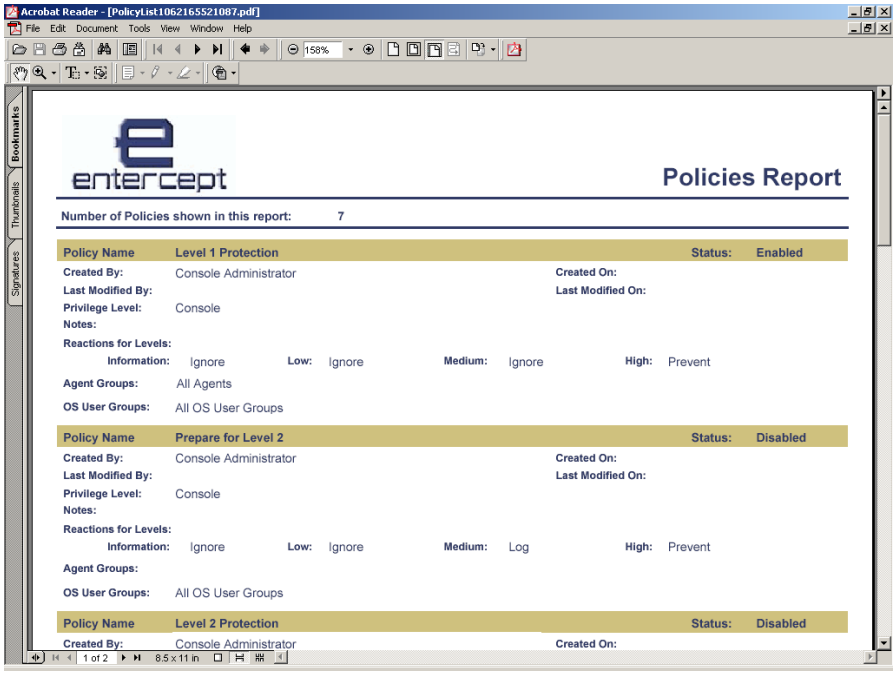

Figure 11 - Entercept: Text report

Several of the reports simply document the system configuration, whilst others provide graphical analyses of security events and reactions.

Available reports include:

- *Security/System Events*
- *Agents/Agent Groups*
- *Exceptions*
- *Policies*
- *Address Book*
- *Notifications*
- *Agent Versions*
- *Signatures*
- *Security Events by Agent/Agent Group*
- *Security Events by Date*
- *Security Events by Reaction*
- *Security Events by User*
- *Security Events by Severity Level*
- *Reactions by Agent/Agent Group*
- *Reactions by Severity Level*
- *Policy Reactions by Severity Level*
- *Agent Uptime*
- *Notification History for Security/System Events*
- *Configuration Preferences*

Most eventualities are covered, which is just as well since Entercept locks down the database and configuration as a security measure, thus preventing additional user-defined reports from being implemented.

Reports can be filtered on a wide variety of parameters (depending on the report subject), including:

- ***Security Level -*** *Info, Low, Medium, High (or all)*
- ***User -*** *Operating system user (one only can be selected, or all)*
- ***Reaction Type -*** *Log, Prevent (or all)*
- ***Signature -*** *One only can be selected (or all)*
- ***Agent -*** *One only can be selected (or all)*
- ***Process -*** *One only can be selected (or all)*
- ***Date Range -*** *From and To dates, or Last X Days (or all)*
- ***Acknowledged By -*** *Entercept user (one only can be selected, or all)*

Each report can be viewed on-screen in PDF or RTF format, and saved as PDF, RTF or Excel files. The biggest problem with the reporting system at the moment is that data is archived every 24 hours to CSV files, after which it is no longer available for reporting.

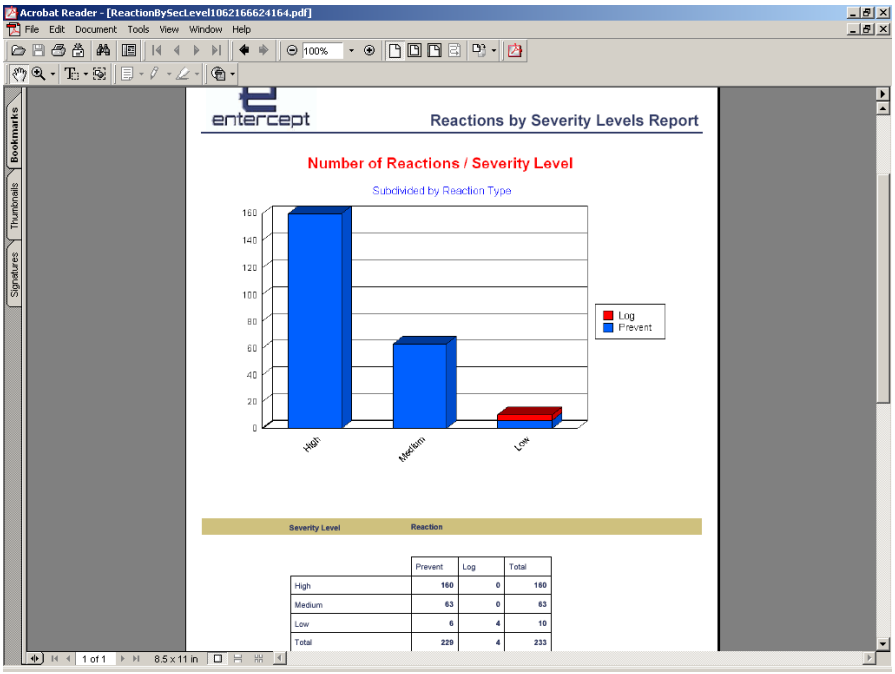

Figure 12 - Entercept: Graphical report

This is not ideal for detailed forensic analysis or long-term trending, although the CSV files could be imported into third party products to provide such capabilities.

Verdict

Performance

In terms of detection and prevention, Entercept performed well, stopping all of the attempted unauthorised access to critical files, directories, registry keys and applications.

It was also very straightforward to create exceptions to allow legitimate operations to proceed where they have been prevented in error.

This level of protection has its price, however, and this comes in the form of an overall lowering of the maximum capacity of the server on which the Agent is installed when under extreme loads. HTTP response times were also affected, but whilst the figures were **numerically** noticeable in our tests, they are in the order of microseconds and are thus unlikely to make a significant impact on the end user experience in most normal deployments.

There was almost no noticeable affect on FTP response times, indicating that the actual impact of the Agent alone is indeed as low as that claimed by Entercept, and that the most significant impact (numerically, rather than real-world) is imposed by the use of the ISAPI filter in Web traffic.

Security Effectiveness

The host-based approach ensures that there are no issues with switched networks or encrypted traffic, and the insertion of the Agent software at kernel level means that the system is capable of protecting the host against both known and unknown attacks with a relatively small impact on the host system.

The provision of Web Server and Database Agents also secures application software within an almost impregnable vault where virtually all attacks will be prevented before hitting the server. Should an attack actually get through, it is then prevented from operating outside the scope of the application server itself. The *Code Red* worm is one example – Entercept users were protected even without the IIS patch in place since the worm was rejected at the HTTP layer before it could deliver its payload.

In all our tests using live exploits, Entercept successfully blocked every attempt to subvert the system, even when evasion techniques were employed. The custom signature capability is much more flexible in the current release than in previous versions of the software, and this allowed us to cover most activities where our requirements fell outside the scope of the built-in rules,

Usability

The Entercept Console is very easy to use, providing excellent Agent update, policy deployment and Agent monitoring capabilities for up to 5000 Agents from a single Management Server. The latest release demonstrates considerable improvements over the last version we tested in our labs.

It is important to realise that this is an Intrusion Prevention System, and not a Host IDS. As such, Entercept does not provide any form of event or system log monitoring capability, and some critical events - such as audit policy or user rights modification - are reported as cryptic registry modifications rather than recognisable alerts.

Forensic analysis is also made more difficult than it should be since it is still possible to select or filter only on a single signature at a time when reporting.

This means that it is often necessary to manually run several reports across several signatures in order to provide a complete analysis of a suspicious event, and it would be nice to see a custom reporting facility that allowed multiple signature coverage in a single report.

Finally, although the user interface has seen significant improvements, there are still some more to be made. In particular, it needs the ability to make block changes to groups of configuration items (such as signatures) in a single operation.

Configuration is straightforward, and there is very little to do to maintain the system on a daily basis. The new three-tier management architecture is more scalable and robust than the previous offering, allowing larger systems to be managed effectively. New features such as the ability to deploy multiple Agents across the network in order to test new versions on a restricted number of hosts and the ability to define and maintain custom signatures in the GUI Console are very welcome, and incredibly useful.

Entercept performed well in all our tests (barring some shortcomings in the reporting capabilities), and this really is one of the few IPS systems we have seen that can be installed, deployed and managed by an administrator with little or no security experience.

Contact Details

Company: Network Associates

E-mail: sales@nai.com

Internet: www.nai.com

Address:
3965 Freedom Circle,
Santa Clara,
CA 95054
USA

Tel: +1 800 338 8754
Fax: +1 408 346 3576

APPENDIX A – TEST RESULTS

The aim of this procedure (based on V1.0 of the NSS Group Host IPS Testing Methodology) is to provide a thorough test of all the main components of a Host IPS product in a controlled and repeatable manner, and in the most "real world" environment which it is possible to simulate in a test lab.

The Test Environment

The network is 100/1000Mbit Ethernet with CAT 5e cabling and a mix of Allied Telesyn AT-9816GB and AT-9812T switches. Software (the "*agent*") is installed on a tri-homed (2xGigabit and 1x100Mbps interfaces) SuperMicro SuperServer 6012P-6 (the "*host*", with dual 1.8GHz Pentium 4 processors and 2GB RAM) on the protected subnet. There is no firewall protecting the target subnet.

Software installed on the host is Windows 2000 SP3 with a standard "out of the box" installation of IIS (including both Web and FTP services). Performance tuning is limited to:

- Moving the "*Performance tuning*" slider to the "*More than 100,000 hits per day*" setting on the Web server
- Disabling Web/FTP logging completely
- Setting "*Unlimited Connections*" on both Web and FTP servers
- Ensuring that both "*Bandwidth Throttling*" and "*Process Throttling*" are disabled on the Web server

Onto the Web server is installed a complete copy of *The NSS Group* Web site - this allows us to ensure that only "real world" data is being requested and served as part of these tests.

Section 1 – Basic Protection Capabilities

The aim of this section is to verify that the agent is capable of detecting a wide range of common attempted intrusions accurately and successfully preventing them without impacting the normal operation of the host.

The latest signature/update pack is acquired from the vendor, and agents are deployed with **all** available signatures enabled (some audit/informational signatures may be disabled).

Note that in some of the following tests it may be necessary to create one or more custom signatures. The product will still be entitled to a **PASS** in these cases providing the signature(s) can be created by the administrator using the tools provided as part of the shipping system. Wherever a test cannot be completed, or can only be completed with the help of a specially constructed software or signature update from the vendor, that test will be deemed a **FAIL**.

Test 1.1 - Critical system files

If an attacker should gain access to the host, he may try to delete system files or replace them with "Trojaned" versions to act as backdoors.

This test determines the ability of the Host IPS agent to prevent access to a range of critical system-related application and data (i.e. password) files without requiring custom signatures.

Test 1.2 - Critical Registry keys

If an attacker should gain access to the host, he may try to amend or delete certain Registry keys in an attempt to subvert the host or change its mode of operation. This test determines the ability of the Host IPS agent to prevent access to a range of critical system-related Registry keys without requiring custom signatures.

Test 1.3 - Specific Registry keys

If an attacker should gain access to the host, he may try to amend or delete certain Registry keys in an attempt to subvert critical applications. This test determines the ability of the Host IPS agent to prevent access to user-specified Registry keys.

Test 1.4 - IPS components

If an attacker should gain access to the host, he may try to remove or disable the IPS agent software. This test determines the ability of the Host IPS agent to prevent access to the main components (application and configuration files) of the Host IPS agent itself without requiring custom signatures.

Test 1.5 - Services/Daemons

If an attacker should gain access to the host, he may try to remove or disable critical system service/daemons. This test determines the ability of the Host IPS agent to prevent unauthorised users from stopping, pausing, modifying or deleting critical system services/daemons.

Test 1.6 - Audit logs

If an attacker should gain access to the host, he may try to amend audit logging operations and/or clear the system audit logs. This test determines the ability of the Host IPS agent to prevent access to the audit logging parameters and record details of, or prevent, clear down of the logs themselves.

Test 1.7 - Protected data files/directories

If an attacker should gain access to the host, he may try to remove, delete or modify certain application data files and/or directories. This test determines the ability of the Host IPS agent to prevent access to user-specified files and directories.

Test 1.8 - Protected applications

If an attacker should gain access to the host, he may try to remove, delete or modify certain application executables to insert Trojan/backdoor software, or may attempt to transfer data out of the protected network using standard applications such as FTP or e-mail.

This test determines the ability of the Host IPS agent to prevent access to user-specified applications (i.e. prevent deletion or modification of PAYROLL.EXE, as well as preventing access to the FTP client on the protected host from certain user accounts).

Test 1.9 - Network services

An attacker may try to exploit certain network services (i.e. Telnet or FTP) on the protected host. This test determines the ability of the Host IPS agent to *prevent* access to specific network services from any remote host, or from a given range of IP addresses, or to *allow* access to specific services from a specific host only.

Test 1.10 - Privilege escalation

If an attacker should gain access to the host, he may try to escalate privileges of certain accounts on that system. This test determines the ability of the Host IPS agent to prevent such privilege escalation (i.e. granting administrator rights to normal user accounts).

Test 1.11 - Buffer overflows

Buffer overflows are a common method of compromising remote hosts. This test determines the ability of the Host IPS agent to prevent buffer overflows from common remote exploits (i.e. live Code Red infection).

Test 1.12 - Web server exploits

Web server vulnerabilities are a common method of compromising remote hosts. This test determines the ability of the Host IPS agent to prevent buffer overflows from common remote exploits (i.e. live Code Red infection, probing with widely available Web scanners, etc.)

Test 1.13 - Exceptions/False Positives

One of the main issues with Host IPS agents is the possibility that they will prevent access to critical system resources by legitimate applications or users. This test determines the ability of the system to quickly and easily define “exceptions” to allow legitimate applications and users to continue to access resources that would otherwise be prevented by any of the aforementioned tests.

Test 1.14 - Reporting

It is important to be able to report on the actions of the Host IPS agent. This test determines the ability of the IPS system to report on all agent actions, whether prevent or log, within a given time frame and subject to user-specified filtration criteria (i.e. for a specific IP address, exploit, agent, etc.)

Section 2 – Performance Under Load

The aim of this section is to determine the impact the Host IPS agent has on the host on which it is installed.

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).

Web traffic is generated by the Spirent Communications Avalanche product, a dual-Gigabit stress testing device that is capable of generating over 1.2Gbps of real Web requests from over 1 million unique users/IP addresses (over 1 million unique open connections, if required) at rates of up to 30,000 connections per second. This level of traffic from even a single Avalanche device is more than capable of saturating the hardware/software platform used for this test.

The Avalanche is connected to a Gigabit switch, to which is also connected the Gigabit port of the SuperMicro Web server. This provides a direct and uninterrupted data path from the Web client to the Web server, ensuring that the network infrastructure could not act as a bottleneck. We realise that most “normal” Web servers would not be expected to serve up to 1Gbps of data, but the aim in this test is to ensure that the network infrastructure does not become a bottleneck.

Test 2.1 - Baseline HTTP - No agent

This test is intended to be representative of “real world ” Web environments. Here we are using a genuine captured browsing session from the NSS Group Web site during which the user accesses our home page, and then moves on to viewing a report on-line resulting in the retrieval of several very large text pages and a large number of accompanying graphics (screen shots). The user then moves on to accessing several smaller pages and menus before completing the session. This session consists of 180 URLs spread over 16 Web pages.

We also use multiple user profiles, each with different think times (60-120 seconds), and limit the traffic to around 100Mbps (an arbitrary value, but we felt that not many public facing Web servers would have Internet connections of more than 100Mbps). In choosing these URLs and bandwidth values, we have, in fact, mirrored exactly the live NSS Group Web server in every respect - including available bandwidth - and thus feel this is an acceptable “real world” scenario.

This particular test is run with a “clean” host - no agent installed, with up to 3500 concurrent users, and 448 transaction per second. Once the test has completed, we record URL response times (*minimum, maximum* and *average*) and page response times (*minimum, maximum* and *average*).

Test 2.2 - HTTP with agent

Test 2.1 is repeated following the installation of the Host IPS agent software. As with the first test, we reach a maximum of 3500 concurrent users, and 448 transaction per second. Once the test has completed, we record URL response times (*minimum, maximum* and *average*) and page response times (*minimum, maximum* and *average*).

This provides us with a direct comparison of response times per-URL and per-page at identical load levels, with and without IPS agent software installed.

Test 2.3 - Baseline FTP - No agent

For those Host IPS agents using ISAPI filters for Web server protection, the performance impact is expected to be significant. The use of a test consisting purely of FTP traffic, therefore, provides us with a useful indication of the overhead imposed by the agent itself rather than the ISAPI filter mechanism.

For this test, we use a genuine FTP session from our Web site, consisting of the retrieval of a single 7.5MB PDF file which is actually one of the NSS Group test reports. As with the HTTP tests, we use multiple user profiles, each with different think times (60-120 seconds), and limit the traffic to around 100Mbps to achieve the same “real world” scenario.

This particular test is run with a “clean” host - no agent installed, with up to 250 concurrent users, and 14 transaction per second. Once the test has completed, we record total number of FTP sessions completed, total FTP data transferred, and the FTP file transfer rate in KB/s (*minimum, maximum* and *average*).

Test 2.4 - FTP with agent

Test 2.3 is repeated following the installation of the Host IPS agent software. As with test 2.3, we reach a maximum of 250 concurrent users, and 14 transaction per second. Once the test has completed, we record total number of FTP sessions completed, total FTP data transferred, and the FTP file transfer rate in KB/s (*minimum, maximum* and *average*).

This provides us with a direct comparison of amount of data transferred and data transfer rates at identical load levels, with and without IPS agent software installed.

Section 3 – Evasion techniques

The aim of this section is to verify that the agent is capable of detecting basic Web exploits when subjected to varying common evasion techniques. Because of the level at which a Host IPS agent operates, the usual evasion techniques which can be applied against Network IDS products should have no effect.

Test 3.1 - Baselines

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

Test 3.2 - Packet Fragmentation and Stream Segmentation

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

- ***Test 3.2.1 -*** *IP fragmentation - ordered 8 byte fragments*
- ***Test 3.2.2 -*** *IP fragmentation - ordered 24 byte fragments*
- ***Test 3.2.3 -*** *IP fragmentation - out of order 8 byte fragments*
- ***Test 3.2.4 -*** *IP fragmentation - ordered 8 byte frag, duplicate last packet*

- ***Test 3.2.5 -*** *IP fragmentation - out of order 8 byte fragments, duplicate last packet*
- ***Test 3.2.6 -*** *IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse*
- ***Test 3.2.7 -*** *IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new)*
- ***Test 3.2.8 -*** *IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old)*
- ***Test 3.2.9 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums*
- ***Test 3.2.10 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags*
- ***Test 3.2.11 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream*
- ***Test 3.2.12 -*** *TCP segmentation - ordered 1 byte segments, duplicate last packet*
- ***Test 3.2.13 -*** *TCP segmentation - ordered 2 byte segments, segment overlap (favour new)*
- ***Test 3.2.14 -*** *TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers*
- ***Test 3.2.15 -*** *TCP segmentation - out of order 1 byte segments*
- ***Test 3.2.16 -*** *TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits*
- ***Test 3.2.17 -*** *TCP segmentation - ordered 1 byte segments, segment overlap (favour new)*
- ***Test 3.2.18 -*** *TCP segmentation - out of order 1 byte segments, PAWS elimination (interleaved duplicate segments with older TCP timestamp options)*
- ***Test 3.2.19 -*** *IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery*

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

Test 3.3 - 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 3.3.1 -*** *URL encoding*
- ***Test 3.3.2 -*** */./ directory insertion*
- ***Test 3.3.3 -*** *Premature URL ending*
- ***Test 3.3.4 -*** *Long URL*
- ***Test 3.3.5 -*** *Fake parameter*
- ***Test 3.3.6 -*** *TAB separation*
- ***Test 3.3.7 -*** *Case sensitivity*
- ***Test 3.3.8 -*** *Windows \ delimiter*
- ***Test 3.3.9 -*** *Session splicing*

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

Entercept 4.1 Test Results

Section 1 - Basic Protection Capabilities

Test	Pass/Fail
Test 1.1.1 - Critical system files	PASS
Test 1.1.2 - Critical Registry keys	PASS
Test 1.1.3 - Specific Registry keys	PASS
Test 1.1.4 - IPS components	PASS
Test 1.1.5 - Services/Daemons	PASS
Test 1.1.6 - Audit logs	PASS
Test 1.1.7 - Protected data files/directories	PASS
Test 1.1.8 - Protected applications	PASS
Test 1.1.9 - Network services	FAIL[1]
Test 1.1.10 - Privilege escalation	PASS
Test 1.1.11 - Buffer overflows	PASS
Test 1.1.12 - Web server exploits	PASS
Test 1.1.13 - Exceptions/False Positives	PASS
Test 1.1.14 - Reporting	PASS
Total Passed	**13 / 14**

Section 2 - Performance Under Load

Test 2.1 – Baseline HTTP - No agent	URL response times (msec)			Page response times (msec)		
	Min	Max	Ave	Min	Max	Ave
Clean host (no agent installed) - 180 URLS/16 Web pages - maximum 3500 concurrent users - 448 transactions per second	<1	<1	<1	<1	<1	<1

Test 2.2 – HTTP with agent	URL response times (msec)			Page response times (msec)		
	Min	Max	Ave	Min	Max	Ave
Agent software installed - 180 URLS/16 Web pages - maximum 3500 concurrent users - 448 transactions per second	<1	<1	<1	<1	<1	<1

Test 2.3 – Baseline FTP - No agent	FTP sessions	FTP data transferred (Kbytes)	FTP data transfer rate (Kbps)		
			Min	Max	Ave
Clean host (no agent installed) - single binary data file (7.5MB) retrieved per transaction - maximum 250 concurrent users - 14 transactions per second	31145	238492288	110771	116475	113768

Test 2.4 – FTP with agent	FTP sessions	FTP data transferred (Kbytes)	FTP data transfer rate (Kbps)		
			Min	Max	Ave
Agent software installed - single binary data file (7.5MB) retrieved per transaction - maximum 250 concurrent users - 14 transactions per second	31065	237862352	109333	117853	113444

Section 3 - Evasion Techniques

Test 3.1 – Evasion Baselines	Detected?
Test 3.1.1 - Test CGI probe (/cgi-bin/test-cgi)	YES
Test 3.1.2 - PHF remote command execution	YES
Test 3.1.3 - FTP CWD root	NO
Test 3.1.4 - Fragroute baseline (test-cgi probe using HEAD)	YES
Test 3.1.5 - Whisker baseline (test-cgi probe using HEAD)	YES
Total	**5 / 5**[2]

Test 3.2 – Packet Fragmentation/Stream Segmentation	Prevented?	Decoded?
Test 3.2.1 - IP fragmentation - ordered 8 byte fragments	YES	YES
Test 3.2.2 - IP fragmentation - ordered 24 byte fragments	YES	YES
Test 3.2.3 - IP fragmentation - out of order 8 byte fragments	YES	YES
Test 3.2.4 - IP fragmentation - ordered 8 byte fragments, duplicate last packet	YES	YES
Test 3.2.5 - IP fragmentation - out of order 8 byte fragments, duplicate last packet	YES	YES
Test 3.2.6 - IP fragmentation - ordered 8 byte fragments, reorder fragments in reverse	YES	YES
Test 3.2.7 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour new)	YES	YES
Test 3.2.8 - IP fragmentation - ordered 16 byte fragments, fragment overlap (favour old)	YES	YES
Test 3.2.9 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with invalid TCP checksums	YES	YES
Test 3.2.10 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with null TCP control flags	YES	YES
Test 3.2.11 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with requests to resync sequence numbers mid-stream	YES	YES
Test 3.2.12 - TCP segmentation - ordered 1 byte segments, duplicate last packet	YES	YES
Test 3.2.13 - TCP segmentation - ordered 2 byte segments, segment overlap (favour new)	YES	YES
Test 3.2.14 - TCP segmentation - ordered 1 byte segments, interleaved duplicate segments with out-of-window sequence numbers	YES	YES
Test 3.2.15 - TCP segmentation - out of order 1 byte segments	YES	YES
Test 3.2.16 - TCP segmentation - out of order 1 byte segments, interleaved duplicate segments with faked retransmits	YES	YES
Test 3.2.17 - TCP segmentation - ordered 1 byte segments, segment overlap (favour new)	YES	YES
Test 3.2.18 - TCP segmentation - out of order 1 byte segments, PAWS elimination (interleaved dup segments with older TCP timestamp options)	YES	YES
Test 3.2.19 - IP fragmentation - out of order 8 byte fragments, interleaved duplicate packets scheduled for later delivery	YES	YES
Total	**19 / 19**	**19 / 19**

Test 3.3 – URL Obfuscation	Prevented?	Decoded?
Test 3.3.1 - URL encoding	YES	YES
Test 3.3.2 - /./ directory insertion	YES	YES
Test 3.3.3 - Premature URL ending	YES	YES
Test 3.3.4 - Long URL	YES	YES
Test 3.3.5 - Fake parameter	YES	YES
Test 3.3.6 - TAB separation (not applicable to IIS)	N/A	N/A
Test 3.3.7 - Case sensitivity	YES	YES
Test 3.3.8 - Windows \ delimiter	YES	YES
Test 3.3.9 - Session splicing	YES	YES
Total	**8 / 8**	**8 / 8**

Notes:

1. Certain network services are covered by built-in signatures (ports 80, 21, 19, 25 and some Trojan ports). However, it is not possible to specify ports in custom signatures to prevent, say, access to a specific server via Telnet.
2. Custom signatures required for all baseline tests. Not possible to create custom signature for FTP exploits.

Section 1: Basic Protection Capabilities

Entercept did well in all our tests, providing the ability to detect most of our activities via the default policies and signatures.

Where our requirements fell outside the scope of the built-in rules, the custom signature capability - which is much more flexible in the current release than in previous versions of the software - allowed us to cover most activities (the only exceptions being custom signature creation for FTP exploits and user-specified network services/ports).

The reporting and forensics capabilities are good, providing most of the reports you would expect from a system of this kind, although some of them required some additional manual work to get exactly the information we needed. With Entercept, it is possible only to select/filter on a single signature at a time, which makes some forensic work difficult since it is often necessary to combine reports using several signatures. The product needs the ability to define custom report templates which include multiple signatures in a single report.

Section 2: Performance Under Load

Entercept stopped all of the attempted unauthorised access to critical files, directories, registry keys and applications, and it was very straightforward to create exceptions to allow legitimate operations to proceed where they have been prevented in error.

As you would expect from an agent incorporating an ISAPI filter, there is some impact on the overall performance of the host Web server once the Agent software has been installed.

The most obvious impact of the Entercept Agent is to reduce the maximum capacity of the server when under extreme loads, as well as to increase the average page and URL response times. However, when the server with and without Agent is compared under the real-world load levels of our tests, the differences - although **numerically** significant in that the average response time is approximately four times greater with the Agent installed than without - are still only in the order of **microseconds**.

Given that all response times, with or without the Agent installed, remain **less than one millisecond** throughout our tests, it is extremely unlikely that the Agent will make a noticeable difference to the user experience.

Finally, the FTP tests show almost no performance degradation between the two tests. This indicates that the actual impact of the Agent alone is indeed as low as that claimed by Entercept, and that the most significant impact (numerically, rather than real-world) is imposed by the use of the ISAPI filter in Web traffic.

Overall, the Entercept software is unlikely to impact negatively on the average user experience, especially when weighed against the benefits of having the Agent installed.

Section 3: Evasion Techniques

None of our evasion techniques had any effect on the detection or prevention capabilities of the Entercept Agent.