Intrusion Detection and Prevention

Concepts & Examples Guide

Release 4.0r3

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Part Number: 093-1840-000, Rev. B

FCC Statement

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. The equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: The equipment described in this manual generates and may radiate radio-frequency energy. If it is not installed in accordance with Juniper Networks' installation instructions, it may cause interference with radio and television reception. This equipment has been tested and found to comply with the limits for a Class B digital device in accordance with the specifications in part 15 of the FCC rules. These specifications are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Consult the dealer or an experienced radio/TV technician for help.
- Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.

Caution: Changes or modifications to this product could void the user's warranty and authority to operate this device.

Disclaimer

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR JUNIPER NETWORKS REPRESENTATIVE FOR A COPY.

Writer: Mark Schlagenhauf

Editor: Lisa Eldridge

Table of Contents

About This Guide

The Juniper Networks Intrusion Detection and Prevention (IDP) system uses multiple methods to detect and prevent network attacks. IDP is designed to reduce false positives to ensure that only actual malicious traffic is detected and stopped.

Organization

This guide is organized into the following sections:

- Chapter 1, “IDP Overview,” provides an in-depth description of the components and functionality that make up the IDP system.
- Chapter 2, “Getting Started,” gets you up and running with your IDP system, helping you add network objects, create and install a security policy, and view log records in the NetScreen-Security Manager (NSM) Log Viewer.
- Chapter 3, “Fine-Tuning Security Policies,” walks you through a step-by-step process of fine-tuning the IDP system to fit your network traffic.
- Chapter 4, “Analyzing Your Network,” details how to analyze your network using the Profiler tools.
- Chapter 5, “Working with Log Records,” includes information and instructions on using the NSM Log Viewer to view log records and analyze attacks.
- Chapter 6, “Using Reports,” includes information about and instructions for using NSM reporting features to view critical network information and generate custom reports.
- Chapter 7, “Creating Effective Security Policies,” provides background information about and directions for maintaining the security policies that protect your networks, including information about terminal rules, notification preferences, and actions.
- Chapter 8, “Managing Attack Objects,” provides information about the signatures and protocol-anomaly attack objects that make up the Attack Object database. This chapter and describes the format used to create custom signatures.
- Chapter 9, “Configuring Sensor Settings,” explains the implicit rules within the IDP system and how to edit their default values on the IDP Sensor.

- Chapter 10, “Using Tagged Interfaces and Virtual Routers,” provides information about using virtual routers with the IDP system and how to pass VLAN traffic through the IDP Sensor.
- Chapter 11, “Implementing High Availability,” details IDP high availability (HA) solutions, including standalone and external HA.
- Appendix A, “Command Line Utilities,” provides information about and instructions for the command line utilities provided with the IDP system.
- Appendix B, “Daemons,” provides a summary of the Daemons that run on the Sensor and handle various processing and monitoring tasks.
- Appendix C, “Common Criteria EAL2 Compliance,” provides information about securing the IDP Sensor to be in compliance with the Common Criteria EAL2 security target.
- Appendix E, “Glossary,” provides definitions for terms that are used in the IDP system and throughout the computer security industry.

Font Conventions

This manual uses variants of a monospace (fixed-width) font in code examples. Table 1 shows each font and its meaning.

Table 1: Font Conventions Used in Code Examples

Font	Meaning
`Regular`	System output
`Italic`	Variables or placeholders in system output
`Bold`	User input
`Bold italic`	Variables or placeholders in user input

Documentation

The *IDP 50, 200, 600, 1100 Installer’s Guide* is shipped in the box with all new IDP Sensors. This guide provides the basic procedures to help you get the IDP system up and running quickly.

Juniper Networks also provides the *IDP Documentation CD* with each Sensor. The documentation CD contains the document set in PDF format. The documentation is also available on the Web at
http://www.juniper.net/techpubs/software/management/idp/

The IDP document set comprises the following:

- *Intrusion Detection and Prevention Concepts & Examples Guide* explains basic concepts of the IDP system and provides examples of how to use the system.
- *IDP 50, 200, 600, 1100 Installer’s Guide* describes the steps to set up and use your IDP system, and hardware components of the IDP appliance.

- *Intrusion Detection and Prevention Upgrade Guide* explains how to upgrade an existing IDP system. For IDP 4.0, the upgrade guide is replaced by the migration guide.
- *IDP-NetScreen-Security Manager Migration Guide* explains how to migrate IDP Sensor Management to NetScreen-Security Manager 2006.1. It also contains a section on the differences between IDP and NSM.

Online Help

The NetScreen-Security Manager user interface (NSM UI) includes *Online Help*. The online help contains a comprehensive set of instructions for using the IDP system, including step-by-step directions for performing common tasks.

Release Notes

Release notes are available on the Web at http://www.juniper.net/techpubs/software/management/idp/

In the *Release Notes*, you will find the latest information about features, changes, known problems, and resolved problems. If the information in the *Release Notes* differs from the information found in the documentation set, follow the *Release Notes*.

Web Access

To view the documentation on the Web, go to:

http://www.juniper.net/techpubs/software/management/idp/

or

http://www.juniper.net/techpubs/software/management/security-manager/

Comments About the Documentation

To obtain technical documentation for any Juniper Networks product, visit www.juniper.net/techpubs/.

For technical support, open a support case using the Case Manager link at http://www.juniper.net/support/ or call 1-888-314-JTAC (within the United States) or 1-408-745-9500 (outside the United States).

If you find any errors or omissions in the following content, please contact us at the e-mail address below:

techpubs-comments@juniper.net

Contacting Customer Support

For technical support, contact Juniper Networks at support@juniper.net or at 1-888-314-JTAC (within the United States) or 408-645-9500 (from outside the United States).

Chapter 1

IDP Overview

This chapter covers the following topics:

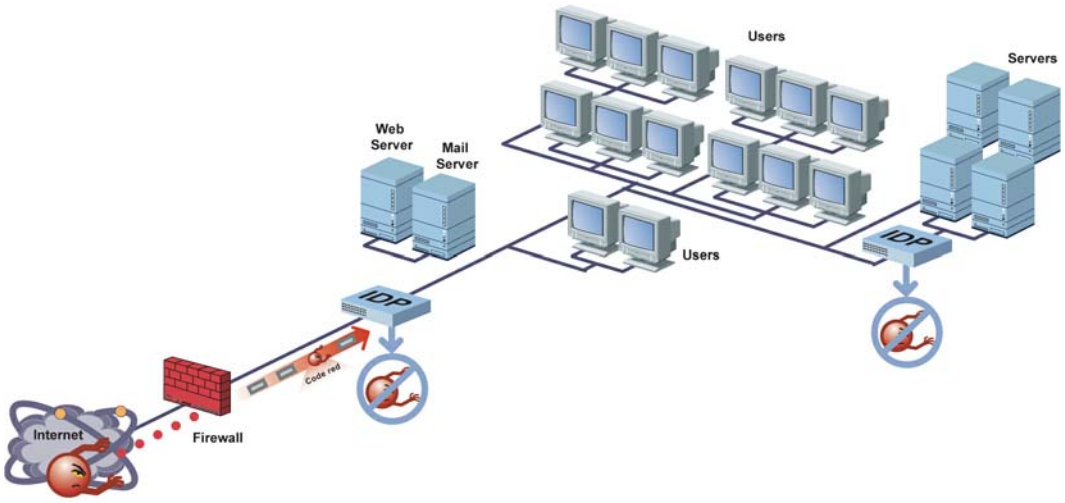
- IDP Sensor
- NetScreen-Security Manager
- User Interface
- High Availability

The Juniper Networks Intrusion Detection and Prevention (IDP) system uses eight detection methods to detect malicious network traffic. It is able to drop attacks to prevent damage to your network and can operate in-line as an active gateway, directly in the path of traffic coming and going on your network:

- When deployed as a passive sniffer, IDP can detect attacks; as an active gateway, IDP can detect attacks and protect your network from them.
- When deployed as an active gateway, IDP prevents detected attacks from reaching their targets. IDP drops malicious packets or connections before they enter your network.

The IDP system's detection and prevention capabilities work against attacks by dropping connections during the attack-detection process, preventing attacks from reaching the target system. You specify what actions the IDP system takes when it detects a particular attack.

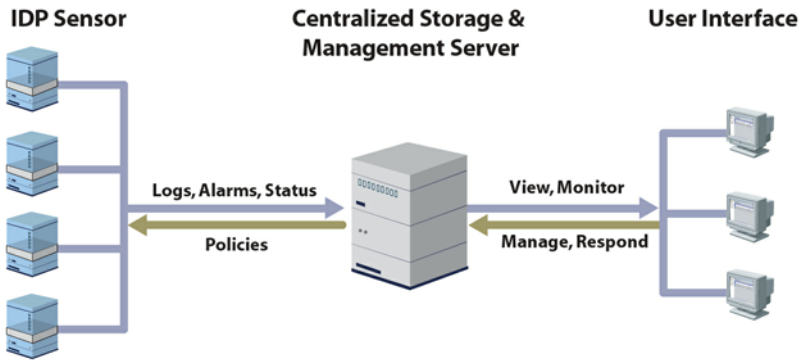
Figure 1: Using IDP in Your Network



The IDP system uses a three-tier architecture that consists of the IDP Sensor, NetScreen-Security Manager (NSM), and the NSM UI.

- IDP Sensors see all network traffic and are the enforcement points that implement security policies.
- NetScreen-Security Manager (NSM) stores and manages all attack objects (including attack signature and protocol anomalies), log information, rulebases, and security policies.
- The NSM user interface (UI) is a graphical interface for interacting with NetScreen-Security Manager. You can use the UI to remotely access and manipulate the information stored in NSM.

Figure 2: Tiers in the IDP System



The following sections detail each tier.

IDP Sensor

The IDP Sensor monitors the network on which the IDP system is installed. The Sensor's primary task is to detect suspicious and anomalous network traffic based on specific rules defined in IDP rulebases. If the Sensor is running in-line, it can also take a predefined action against malicious traffic.

You can deploy IDP Sensors as active gateways or passive sniffers:

- An active-gateway Sensor sits between the network and a firewall, or a network and a DMZ, and takes an active role in protecting the network. When it detects intrusions or attacks defined by security policies, the Sensor can drop, block, or ignore the suspicious connection—or drop only the suspicious packets. Because an active-gateway Sensor can take action against the attack, it can prevent the attack.

- A passive sniffer Sensor connects to the HUB or SPAN port of a switch and sniffs the network traffic as it passes by (you can also use a network TAP). The Sensor monitors network traffic, records security events, and can create alerts for attacks. However, because a sniffer Sensor cannot take action against the attack, it cannot prevent it.

Use the Appliance Configuration Manager (ACM), a web-based tool, to configure an IDP appliance for your network. The ACM leads you through the configuration of the root and admin passwords on the IDP Sensor, network and system settings, and HA configuration. For more information about using the ACM to configure the IDP appliance, see the *Installer's Guide* for the IDP 50, 200, 600, and 1100 appliances or the *Quick Start Guide* for the IDP 10, 100, 500, or 1000 appliances.

NetScreen-Security Manager

In previous releases of IDP, Sensors were managed with the IDP Management Server. As of IDP 4.0, Sensors are managed using the NetScreen-Security Manager (NSM). NSM allows the centralized administration of Juniper Networks firewalls, IDP Sensors, and ISG devices with one user interface.

NSM centralizes the logging, reporting, data, and security-policy management for the IDP system. All objects, security policies, and log records are stored in databases in NSM and are administered using the NSM user interface (UI). NSM communication with the other two tiers of the IDP system (the Sensor and UI) is encrypted and authenticated to provide an additional level of security.

NetScreen-Security Manager performs the following functions:

- Centralizes management of enterprise security policies

- Consolidates logs from different Sensors in a single repository

- Simplifies signature and protocol-anomaly attack-object management

- Enables multiple users to interact with the Sensors

- Manages multiple Sensors

For additional information about NSM, refer to the *NetScreen-Security Manager Administrator's Guide.*

User Interface

The NetScreen-Security Manager user interface (NSM UI) provides a powerful, graphical environment for centrally managing IDP. The NSM UI is a java-based software application that can be installed on multiple computers on your network.

You can configure the UI with your own preference. IDP stores user preferences and custom Log Viewer views in the central database so that they remain consistent when you access them from different client machines. The UI also provides online help.

For more information, refer to the *NetScreen-Security Manager Administrator's Guide.*

High Availability

You can also deploy the IDP system in a high availability (HA) implementation to provide failure protection, either in a standalone configuration or using external hardware. In an HA configuration, two Sensors are joined together in an HA cluster to provide failure protection and/or load balancing. State information is shared between Sensors in the cluster using state-sync interfaces and dedicated network interface cards (NICs).

HA configurations are available for all deployment modes except Sniffer mode:

- Bridge and Transparent modes. Sensors running in Bridge or Transparent mode support the use of external HA hardware for HA and load balancing. You can also configure Spanning Tree Protocol (STP) on Sensors running in Bridge or Transparent mode.

NOTE: You must configure support for STP manually through the command line utilities.

- Proxy-ARP mode. Sensors running in Proxy-ARP mode support the use of standalone HA, using load balancing or hot standby for failure protection:
  - *Load Balancing*. All Sensors in the cluster share network traffic equally. If a Sensor fails, network traffic is redirected to the other Sensors in the cluster.
  - *Hot Standby.* A primary Sensor handles all network traffic while a secondary Sensor stands by. If the primary Sensor fails, network traffic is redirected to the secondary Sensor.
- Router modes. Sensors running in Router mode support the use of standalone HA or external HA hardware for high availability (HA). Standalone and external HA offer load balancing or hot standby for failure protection, as detailed above.

For more information about Juniper Networks HA solutions, see "Implementing High Availability" on page 141.

Chapter 2

Getting Started

This chapter covers the following topics:

- Creating Sensor Security Devices
- Running the Profiler
- Configuring a Security Policy
- Viewing Log Records

To begin using the Intrusion and Detection Prevention (IDP) system, you first add the IDP Sensors as security devices in the NetScreen-Security Manager user interface (NSM UI), run the Profiler to analyze your internal network, and then select and install a security policy that describes how you want your Sensor to protect your network.

The Profiler is a network-analysis tool that helps you learn about your internal network, enabling you to create effective security policies and minimize unnecessary log records. After you configure the Profiler, it automatically learns about your internal network and its elements. You can use this information to create a security policy that accurately detects the attacks to which your network is vulnerable.

Additionally, NSM contains several security-policy templates you can quickly customize to your network. If you are unfamiliar with NSM or IDP, or are new to policy-based management, using a template can help get you up and running quickly.

NOTE: To start with a blank security policy, see “Creating Effective Security Policies” on page 53 for information about designing custom rules.

After you install a security policy on your IDP Sensors, they immediately begin monitoring your network traffic. If directed by the security policy, the Sensor generates a log record for each security event, which can be an attack against your network, a protocol anomaly, or a simple login attempt.

Log records appear in the Log Viewer, where you can analyze them to determine the effectiveness of your current security policy. Log records are also a valuable insight into your network traffic: You can see where traffic is coming from, where traffic is going to, and any malicious content the traffic contains.

Creating Sensor Security Devices

When you initially deploy NSM on your network and open the UI for the first time, there are three types of objects you should become familiar with:

- Security devices. IDP Sensors, once they are added, appear in the UI as security devices.
- Attack objects. NSM includes a database of IDP attack objects that represent known and unknown attack patterns.
- Service objects. NSM includes a database of service objects that represent standard network services such as TCP, UDP, RPC, and ICMP.

NOTE: If you have an existing IDP Management Server system installed, refer to the *IDP-NetScreen-Security Manager Migration Guide* for instructions on migrating your current configuration and data.

No default address objects appear because the Address Object database is empty. You must use the Object Editor to create address objects such as:

- Servers
- Individual host machines
- Networks

All address objects that you create are added to your Address Object database. You can also use the Object Editor to add service objects that represent unique services on your network, attack objects customized to represent attacks unique to your network, and groups of object types.

For now, you only need to create a security device entry for each IDP Sensor on your network; later, you should also create address objects for the network components you want to protect. You can use the method described below to create address objects for your network components, or you can use the shortcut described in “Creating Address Objects” on page 28 to quickly add address objects from the Log Viewer.

To add an existing IDP Sensor as a security device, perform the following steps:

1. In the navigation tree, select Device Manager > Security Devices.
2. Click the + button in the upper left corner and select Device.
3. In the New Device dialog, enter a name for the device.
4. Leave Device Is Reachable checked and click Next.
5. Enter the IP address for the Sensor on the displayed page.
6. For Admin User Name, enter admin.

7. Enter the admin and root passwords in the appropriate fields. (Both passwords are set using the ACM. Refer to the *IDP Installer's Guide*.)
8. Leave the connection information at the default settings and click Next.
9. View, and confirm, the displayed SSH key, then click Next.
10. After a few moments, Sensor information is displayed. Click Finish to close the wizard. The Sensor is displayed in the Security Devices window.
11. Right-click on the device, then select Import Device. Sensor configuration information downloads into the NetScreen-Security Manager database.

Object Editor Tool Tips

To get specific information about an object, place your cursor over the object. For example, placing your cursor over a security device provides the following information about the device:

- Name
- Device type
- Operating system
- IP address
- NSM domain
- Connection state
- Configuration state
- Validation status

Running the Profiler

You next use the Profiler to configure each Sensor on your network to automatically learn about your internal network and its elements, which include:

- Hosts
- Peers
- Ports (non-IP protocols, TCP/UDP ports, RPC programs)
- Data from layer 7 that uniquely identifies hosts, operating systems, applications, commands, users, and filenames

As your Sensors learn, they collect and store data about each unique event on your network, creating a snapshot of the devices and activity on your network. You can then view this data in the NetScreen-Security Manager Profiler and Security Explorer. If you are using multiple Sensors to collect data, the Profiler synchronizes

all data across those Sensors and eliminates redundant information, enabling you to view only those events that are unique. For details about the Profiler and Security Explorer, refer to the discussion about analyzing your network in the *NetScreen-Security Manager Administrator's Guide*.

To run the Profiler, perform the following steps:

1. Open Device Manager > Security Devices and double-click the Sensor you selected. Select Profiler Settings in the Device window:
   - In the Tracked Hosts tab, select the network objects you want the Profiler to collect detailed information about. These should be the objects that represent your internal network.
   - In the Exclude List tab, select any individual objects you want the Profiler to exclude from the groups you selected on the Tracked Hosts tab.
   - In the Context to Profile tab, select Contexts to Profile to profile all contexts.
   - In the Alert and General tabs, leave the default settings.
2. Click OK.
3. Select Devices > IDP Profiler > Start Profiler from the NSM menu bar.
4. Select the Sensor that will run the Profiler. You can select more than one Sensor.
5. A job information dialog appears to tell you about Profiler startup progress.
6. Select Security Monitor > Profiler to see the data you are collecting.

Configuring a Security Policy

Security policies, rulebases, and rules are the core of the IDP system:

- Rules are basic instructions that direct the behavior of the IDP Sensor. You can have multiple rules; rules are organized into rulebases.
- Rulebases are collections of rules that use the same detection method to detect and prevent network intrusions. The IDP rulebases combine to form your security policy.
- Security policies are collections of rules and rulebases. You can have multiple security policies stored in the IDP system, but an IDP Sensor can use only one security policy at a time.

Installing a security policy on an IDP Sensor gives the Sensor a set of instructions about the actions you want the Sensor to take in response to specific network situations. You can install the same security policy on multiple IDP Sensors, or you can install unique security policies on each Sensor in your network.

Using a Security-Policy Template

IDP includes several templates that you can use to create a security policy. The getting_started template is specifically designed to help you first monitor your network traffic, then edit your security policy based on the log records you receive, and finally actively prevent intrusions by closing or dropping malicious connections.

NOTE: Only IDP Sensors that are running in-line can close or drop connections. IDP Sensors in Sniffer mode cannot affect connections.

Creating and Installing a Security Policy

The following sections detail how to create a new security policy using the getting_started template and how to install that policy on your IDP Sensors. “Fine-Tuning Security Policies” on page 23 gives you step-by-step instructions on further customizing the getting_started template to your network.

To create and install a security policy, perform the following steps:

1. In the navigation tree, select Security Policies, then click the + button in the toolbar to create a new security policy. The New Security Policy dialog box appears.
2. In the Name box, enter the name for the security policy. You can use letters, numbers, "-" or "_". Each security policy name must be unique.
3. In the Description box, enter a description, then click Next.
4. Select Create new policy.
5. Deselect Firewall/VPN device.
6. Select Stand Alone IDP Device.
7. Click Next.
8. Select Use IDP Template.
9. In the Name menu, select getting_started, then click Next.
10. Select one or more Sensors that will run the policy. (If you’re not sure, you can assign the policy to a Sensor later.) Click Next.
11. Click Finish.
12. Select the security-policy name to display the security policy in the main display area.
13. Select the IDP tab to display the IDP rulebase.
14. In the Install On column, right-click and select Select Target. Select the Sensors on which you want to install the policy. Click OK. This updates the NSM database for the Sensor, but does not actually push the policy to the Sensor.

15. Select Device Manager > Security Devices. Right-click the Sensors you want to push the policy to and select Update Device.

16. Leave the defaults settings in the displayed dialog and click OK.

17. A job information dialog displays as NSM pushes the policy to the Sensor.

Viewing Log Records

When a Sensor detects a match between the network traffic it is processing and one or more of the rules in the installed security policy, it can perform a user-specified action such as logging the event, closing the matching connection, or dropping the matching packets.

If you install a security policy that uses the getting_started template, your Sensors begin generating log records for each match they detect. These log records are stored in the log file on the IDP Management Server and appear in the Log Viewer component of the UI.

NOTE: If you are not receiving log records, check your Sensor connections and configuration and ensure that a policy is installed.

Log records contain a lot of important data about the activity on your network, but you might not want to see all the information at one time. You can use the Log Viewer to manipulate, analyze, and export the information contained in your log records so you see only the information that interests you. For example, you can use the Log Manager to perform the following tasks:

- View complete, summarized, or detailed information for each log record.
- Show, hide, or move columns; and filter data by column headings. Create a new instance of the Log Viewer that shows only your filters and column settings.
- Set flags on log records to indicate a specific priority or action.
- Launch the Packet Viewer.
- Print or export log record data.

For more details on the Log Viewer component, refer to the information on reporting in the *NetScreen-Security Manager Administrator's Guide*. Or, if you are working with the getting_started security-policy template, see "Fine-Tuning Security Policies" on page 23 to customize the policy to your network.

Chapter 3

Fine-Tuning Security Policies

This chapter covers the following topics:

- Understanding the Fine-Tuning Process
- Investigating Log Records
- Identifying and Eliminating False Positives
- Identifying and Responding to Real Attacks
- Preventing Attacks with Multi-Detection Methods

After you have set up your Sensors and installed a security policy based on the getting_started template, you are ready to begin fine-tuning. Fine-tuning is the process of learning to recognize the information NetScreen-Security Manager (NSM) is giving you and how to tell NSM what you want to do about it, including:

- Comprehending the information in log records, the Log Viewer, the Log Investigator, reports, and the Profiler
- Understanding how NSM represents your network elements, including servers and other machines you want to protect
- Telling NSM which events you want log records created for and eliminating the noise caused by false positives and other non-meaningful network events
- Recognizing actual attacks and telling your Sensors to respond to attacks when it detects them on your network

Understanding the Fine-Tuning Process

Fine-tuning your IDP system is a step-by-step iterative process. Typically, you begin fine-tuning by examining your log records through the Log Viewer, reports, the Log Investigator, or the Profiler, then by making changes to your security policy. When new log records appear that reflect those changes, you again examine the log records to see the results.

This chapter describes how to fine-tune your IDP system. However, because all networks are different, the steps recommended here are only guidelines. You might need to perform some tasks more than once to get the results you want.

Fine-tuning consists of the following four steps:

1. Investigating network traffic patterns through log records, reports, the Log Investigator, and the Profiler, then adding your hosts to NSM as address objects
2. Identifying and eliminating false positives
3. Identifying and responding to real attacks against your network
4. Using multiple detection methods in other rulebases to improve the accuracy of IDP attack detection

Before you begin fine-tuning the IDP system, however, you should understand how the IDP system operates. The following sections provide an overview of how the UI components work together to help you explore and respond to network activity.

Creating Rules for IDP Sensors

The NSM UI contains components for editing rules in rulebases and for viewing the log records that the rules generate. You tell the IDP system how to react to network traffic by creating rules in one or more of the IDP rulebases. These IDP rulebases combine to create your security policy, which you then install on your IDP Sensors.

As network traffic flows through your Sensors, the installed security policy tells the Sensors which network traffic to monitor and possibly take action against. You create rules by specifying:

- Patterns or circumstances that you want to match. NSM comes preinstalled with an Attack Object database, a large collection of information about known attack patterns and other suspicious traffic. In addition, Juniper Networks provides periodic updates to this database, which you can download. This database contains two types of attack objects:
  - Signature attack objects represent known attack patterns, such as buffer overflows and email viruses.
  - Protocol-anomaly attack objects represent patterns or circumstances that indicate suspicious traffic, such as illegal or ambiguous packets.

  You can select individual patterns or groups of patterns from both types of attack objects to use in your rules. You can also create your own pattern and add it to the attack object database. For more information about attack objects, see “Viewing and Editing Signature Attack Objects” on page 95.

- Actions to take when a match is found. The IDP contains a default set of actions that the Sensor can perform against network traffic that matches the specified pattern or circumstance. These actions include dropping or ignoring the attacking network connection or packets, or simply logging the match as a security event and creating a log record that appears in the Log Viewer.

  For more information about specifying actions in your rules, see "Setting Actions" on page 67.

Using Log Records

When an IDP rule contains logging or log packets as a notification action, the Sensor creates a log record for events that match that rule. To see what is happening on your network, view the log records using the following three UI components:

- Use the Log Viewer to view complete, summarized, or detailed log-record information. For details on the Log Viewer, refer to the discussion about logging in the *NetScreen-Security Manager Administrator's Guide.*

- Use the Reports component to view reports generated by log-record information. For details on working with reports, refer to the discussion about reporting in the *NetScreen-Security Manager Administrator's Guide.*

- Use the Log Investigator to correlate log record data.

From these components, you can quickly navigate back to the Security Policy Editor, where you can make changes to the security-policy rules in the IDP rulebases in order to change IDP behavior.

Log records are generated when a rule in the IDP rulebases contains the action Logging and a network traffic event matches the Match portion of the rule.

Logs contain a great deal of information about what is occurring on your network. The getting_started security-policy template logs for all rules in the Main rulebase. To examine those log records, open the Log Viewer.

Because the getting_started security-policy rules were very broadly designed to match *any* destination and *any* source on your network for several services, a great many logs may have been generated. One way of gathering information quickly from large numbers of logs is to view summary information in the Log Investigator or in the IDP Reports component. Refer to the NSM *Online Help* for more information about using those UI components.

Now that you are familiar with these basic concepts of the IDP system, you are ready to begin the four-step process of fine-tuning IDP.

Investigating Log Records

In this step, you determine which devices on your network are important (and therefore attractive to attackers), and how to modify the getting_started template to detect attacks against these devices.

The getting_started security-policy template contains rules that are specifically designed to demonstrate the iterative process of fine-tuning your IDP system. Your daily logs and basic reports can help you quickly identify the devices on your network you should be protecting.

You can view log records through the Log Viewer, the Log Investigator, or the IDP Reports component to get the most complete picture of the activity on your network:

- The Log Viewer displays complete, summarized, or detailed log-record information about attacks against your network:
  - To see your daily logs, click the Log Viewer in the navigation tree.
  - The default view is all log records. To see a filtered view, select a predefined view from the list under the Log Viewer in the navigation tree.
  - You can also create your own filtered views. For details on designing and saving Log Viewer views, refer to the discussion about reporting in the *NetScreen-Security Manager Administrator's Guide.*
- The Reports component displays reports generated by log-record information:
  - To see a report created from your log records, click Reports Manager in the navigation tree, then click DI/IDP Reports. Select a report from the list that appears.
  - Default reports are automatically generated from the information in your log records.
  - You can define your own custom reports based on any combination of log data and filtering you want to display. For details on designing custom reports, refer to the discussion about reporting in the *NetScreen-Security Manager Administrator's Guide.*
- The Log Investigator correlates log-record data to help you identify attack patterns over time. Use the Log Investigator to quickly investigate suspicious activity so you can determine whether an attack exists

.

NOTE: For more information about the NSM UI, refer to the NSM *Online Help*.

Identifying Important Sources and Destinations

All attacks have a source (where the attack is coming from) and a destination (the target of the attack). An IDP log record contains the source and destination of the attack.

Log Viewer

To see attack source and destination in the Log Viewer, look at the Source Address and Destination Address columns:

- The source address of an attack tells you the IP address of the computer that generated the attack. This can be a computer on the external network (an outside attack) or a computer on your internal network (an internal attack).
- The destination address of an attack tells you the IP address of the computer that the attack targeted. This can be any computer or device on your network.

Each Log Viewer column contains several different IP addresses that each represent a host computer. Some hosts are devices on your network (your internal hosts) and others are devices outside your network (external hosts).

An important part of fine-tuning your IDP system is determining which internal hosts you want to protect. An important source or destination might be an IP address that is receiving more attacks than other hosts, or it might be the IP address of a critical device on your network, such as a mailserver.

Reports

The best method for quickly identifying the most frequently targeted hosts is to view a report. In the navigation tree, click Reports, followed by Predefined Reports, and then click the title of the report you want to view.

- To see important sources, click one of the Top Attackers reports.
- To see important destinations, click one of the Top Targets reports.

Each report displays a graph and table indicating the host IP addresses and their number of attacks. Review the Top Attackers and Top Targets reports to determine your most important sources and destinations.

Creating Address Objects

After you have identified your important internal network components, you should create address objects for them.

Creating your own custom address objects can help you perform the following tasks:

- Eliminate unimportant log records. After you create address objects for your network components, you can use the objects to designate specific sources and destinations in your security policy. The Sensor then only generates log records for matches detected in the network traffic between the specified source and destination. This eliminates unimportant log records, reducing the number of log records.

- Make log records easier to read. An address object contains the IP address and host name of the computer it represents, enabling the log records that include an address object as a source or destination to display the host name instead of the IP address. This can make your logs much easier to read, helping you to focus on detecting attacks against the devices you care about.

Using Address Objects

After you create address objects for your important hosts, you can tell IDP to detect attacks that originate from specific sources and that target specific destinations. To detect attacks that target your network, specify your address objects as the Destination of an attack. To detect attacks that emanate from your network, specify your address objects as the Source of an attack.

To edit the source and destination addresses in your security policy, perform the following steps:

1. In the Destination column of a rule in the IDP rulebase, replace any with the address objects that represent your destination hosts.

2. Verify and install the edited security policy on your Sensor.

3. Review your log records in the Log Viewer. For attacks that target your address objects, IDP displays the host name as the destination.

Identifying and Eliminating False Positives

A false positive is a log record that reflects an event on your network that you are not concerned about and do not want to see in your logs. In the IDP system, a false positive is any log record that looks like an attack but is actually a harmless event. This log record is generated because IDP has detected a valid match between an attack object in your security policy and your network traffic, but the event is not a threat.

Common causes of false positives are as follows:

- False alarms caused by benign network traffic
- False alarms caused by nonstandard software configurations
- Actual attacks that are irrelevant to your situation
  - Attacks against services your network does not support
  - Attacks that you are already protected against via patches, and so on

While false positives in your logs are harmless, they can distract you from real attacks, which can be frustrating. Identifying and removing false positives from your logs helps you focus on attacks that actually threaten your network, reduces the number of log records, and increases IDP performance.

Identifying False Positives

Most false positives generated from benign traffic are easy to identify because they look very different from actual attacks.

Attackers have a specific agenda. They want to compromise your network with a minimal amount of activity in the shortest possible time. If an attack is not working against your network, attackers can change tactics and try a new attack as they attempt to discover a way into your system. False positives, however, have no agenda, no organization, and do not change over time, making them easy to identify in logs.

In this step, you learn some ways of identifying false positives and how to eliminate the noise they cause in your logs.

To identify false positives in logs, use the information in Table 2 to determine if the attack is a false positive.

Table 2: Identifying False Positives from Real Attacks

False Positive	Real Attack
Attack comes from a trusted or known source address.	Real attacks often come from sources you do not know or trust. Attackers typically want to mask their identity by attacking your network from an unknown IP address.
Attack is frequent and creates large numbers of log records.	Real attacks can be stealthy. Multiple log records for the same attack are highly visible in your logs. Attackers typically do not announce their presence by using an attack that generates dozens or hundreds of log records.
Attack spans a long time period.	Generally, real attacks are brief. Attackers typically strike swiftly to minimize their chances of being discovered. An exception to this is a focused stealth attack that spans a long period of time, specifically for the purpose of appearing innocent.
Attack is unrelated to other attacks in the same time period.	Real attacks are coordinated. Attackers typically attempt to discover security holes in your network by attacking a machine using different methods.

Keep in mind that false positives are specific to your network and the traffic that flows on it. Attack objects that generate false positives for your network might detect real attacks on another network—or different network traffic.

Eliminating False Positives from Benign Traffic

False positives can generate irrelevant log records. In some cases, you might want to eliminate all log records for a particular type of attack that is irrelevant throughout your network. In other cases, you might want to continue logging a particular type of attack but ignore the attack for a particular combination of source and destination where the attack is irrelevant.

To eliminate a false positive that is caused by a rule in the IDP rulebase that is irrelevant for your entire network, you must remove the associated attack object from the rule that triggered the false positive.

To remove the matching attack object from a rule in the IDP rulebase, perform the following steps:

1. In the Subcategory column of the Log Viewer, right-click the log record of the false positive and select Exempt.

   NSM jumps to the Exempt rulebase for the policy and adds a new rule. The new rule creates an exemption for that attack for the source-destination combination found in the log record.

2. Edit the new rule if you want to add additional entries to the source, destination, or attack fields.

3. Close the policy window and go back to the log record.

4. Repeat for all false positives you do not want IDP to continue detecting.

5. Verify and install the edited security policy on your Sensor.

6. After some time has passed, review your logs in the Log Viewer. No log records for exempted source/destination/attack combination appear.

Eliminating False Positives from Internal Software

The software you run on your network can occasionally produce network traffic that matches attack objects in your security policy. You can tell IDP to *stop* detecting matches for that attack object in traffic that emanates from your internal network, but to *continue* detecting matches in traffic that emanates from the external network.

You can achieve this result either by creating an Exempt rule for traffic with a source address in your internal network, as described in the previous section, or by creating a separate rule to ignore the internal traffic.

To create a rule that ignores internal traffic when detecting an attack-object match, perform the following steps:

1. In the Security Policy Editor, select the rule that contains the matching attack object.
2. Right-click the rule and select Edit > Copy.
3. In the same rule, right-click and select Edit > Paste > Before.
4. Right-click the Attacks column (old rule) and remove the matching attack object.
5. Right-click the Attacks column (new rule) and remove all attack objects *except* the matching attack object.
6. Right-click the new rule's Source column and add your internal address objects.
7. One at a time, click each address object in the Source column and choose Negate.
8. Verify and install the edited security policy on your Sensor, then review your logs in the Log Viewer.

IDP continues detecting attacks that match the attack object in network traffic that originates outside your network, but it no longer monitors internal network traffic.

Identifying Irrelevant Attacks

All operating systems and software contain potential vulnerabilities, especially when they are first released or when new versions are first available. That means that the components on your network are inherently vulnerable. However, it is unlikely that you need to protect against all vulnerabilities for all computer devices and applications. Attacks that target vulnerabilities you do not have can be considered another form of false positive.

For example, you are not vulnerable to attacks targeting hardware or software you do not use or a software version you have not installed. Because attackers may not know which hardware and software are on your network, they may use broad-range, untargeted attacks against your network to search for vulnerabilities. These attacks are a nuisance but not a threat. You can modify your IDP rules so you do not even see them.

Untargeted attacks look different from serious threats, just as benign false positives look different from real attacks. Use the following steps to determine if log records that look like attacks are events you should be concerned about. By identifying and eliminating irrelevant attacks:

- You can reduce the number of log records and increase IDP performance.
- You can isolate log records for harmless attacks.
- You can focus on log records for attacks to which you are actually vulnerable.

1. Open the Log Viewer and select a log record. View the Summary panel to see the attack description.
2. Determine if the attack is relevant by reviewing the information about affected systems and comparing it with what you know about your network.

Table 3: Identifying Irrelevant Attacks

Irrelevant Attacks	Relevant Attacks
Attack targets hardware you do not use. Example: Attacks that exploit Cisco routers do not target Lucent routers.	Attack attempts to exploit vulnerabilities in the hardware you use in your network.
Attack targets software you do not use. Example: Attacks that exploit Microsoft IIS webservers do not target Apache webservers.	Attack attempts to exploit vulnerabilities in the software running on your network.
Attack targets software versions you do not use.	Attack attempts to exploit vulnerabilities in the software versions running on your network.

3. If irrelevant, remove the matching attack object from the rule that triggered the log record, or monitor the attack object using a custom severity.

Monitoring Irrelevant Attacks

Although irrelevant attacks are not immediate threats to your network, you should still view them as important. Evidence of these attacks in your daily logs might indicate that attackers are attempting to guess what hardware and software your network uses. If they are persistent, eventually they might guess correctly and succeed in attacking your network with a more dangerous, targeted attack. Keep this in mind when deciding how to manage your untargeted attacks.

To create a rule that monitors irrelevant attacks, perform the following steps:

1. In the Security Policy Editor, select the rule that contains the matching attack object.
2. Right-click the rule and select Edit > Copy.
3. In the same rule, right-click and select Edit > Paste > Before.
4. Remove the matching attack object from the original rule.
5. Remove all attack objects from the new rule except the matching attack object.
6. In the severity column of the new rule, right-click and change the severity setting to Warning.
7. Verify and install the edited security policy on your Sensor.
8. Review your logs in the Log Viewer. All log records for the matching attack object now filter into the low severity view.

This approach adds more rules to your security policy to give you more granular control. It also organizes your log records by isolating irrelevant attacks and suspected false positives, making it easier to identify real attacks.

Identifying and Responding to Real Attacks

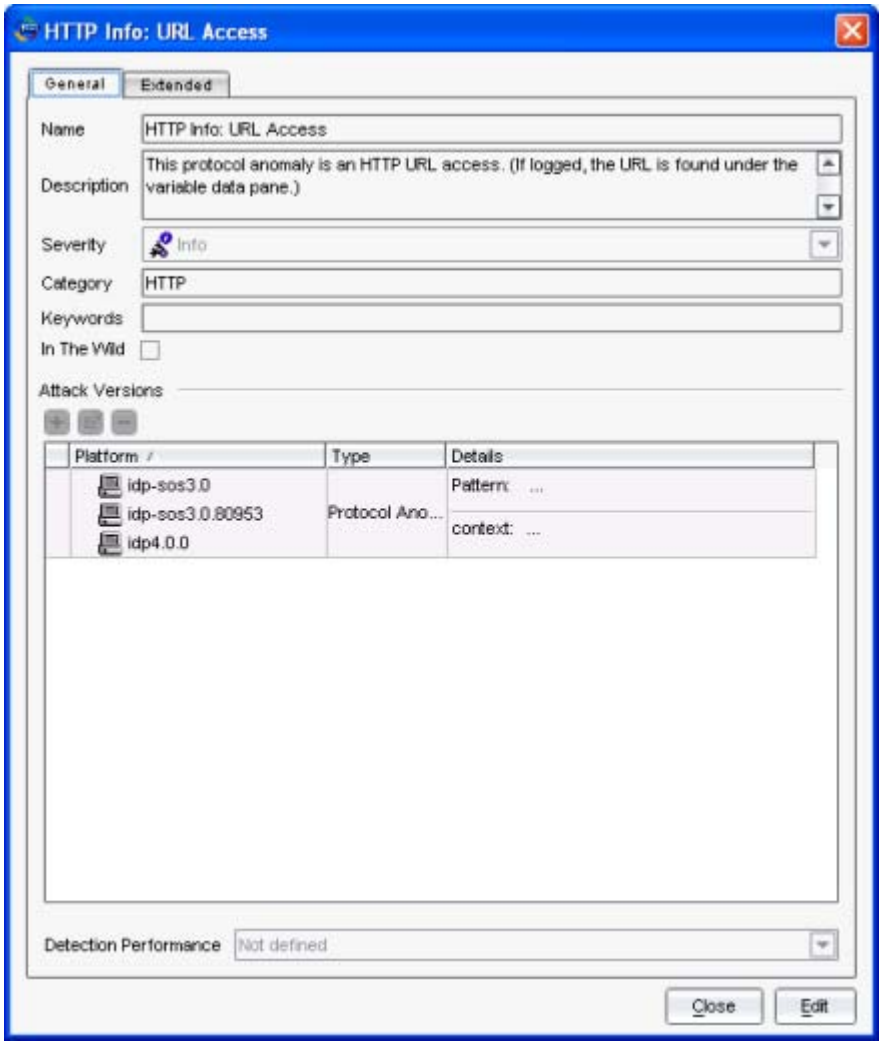
In this step, you investigate your log records for real attacks and set alerts that are triggered when those attacks occur. You might have already discovered several real attacks as you identified untargeted attacks and false positives. Now you can look closer at those attacks, verify their authenticity, and decide how to respond.

Identifying real attacks can seem like more of an art than a science. You can use the integrated security incident management features of IDP to analyze an attack and plan your response.

Viewing Attacks in the Attack Object Editor

As you explore your log records in the Log Viewer, double-click on the Subcategory field of a log record to view a description of the attack object that triggered the log record.

Figure 3: Attack Object Editor



The Attack Object Editor contains important information about the attack and defines the attack-severity level. Severity is the property of an attack object that indicates how serious this threat is to your network.

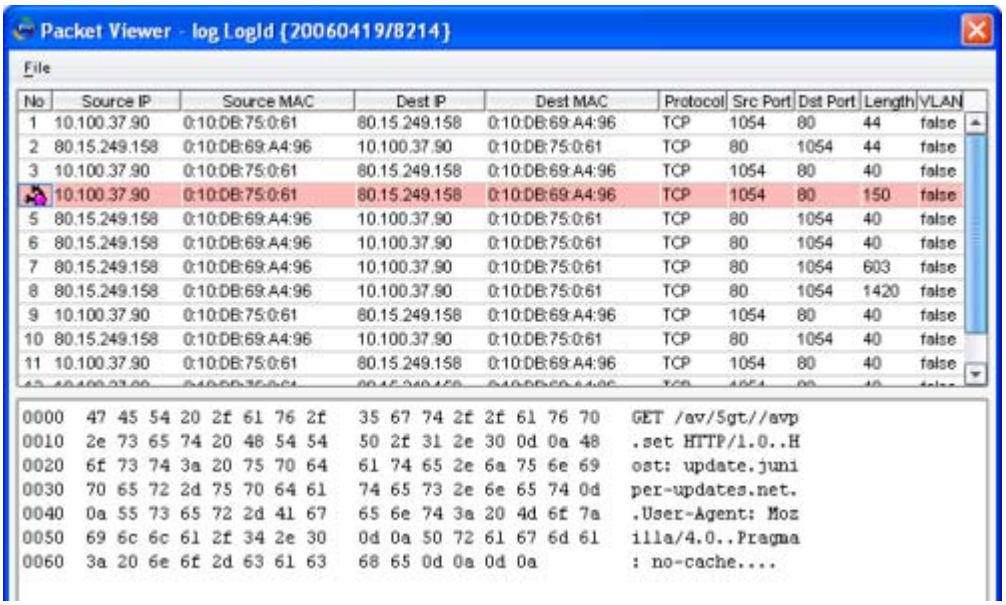
The Attack Object Editor also contains an Extended tab that displays detailed information about the attack, if available, including links to outside information about the attack, such as the attack CVE number and description, the products and vendors affected, patch availability, and workarounds. Using the attack information, the attack severity level, and external data about the attack, you can determine whether the event is a real attack.

NOTE: The Attack Object Editor is also where you create and modify custom attack objects. For details, see “Managing Attack Objects” on page 93.

Viewing Attacks in the Packet Viewer

You can also use the packet viewer to identify real attacks. As you explore your log records in the Log Viewer, right-click on the log record, then select Show > Packet Data to see the exact packet data that triggered the log record. The packet that triggered the alert is highlighted in red in Figure 4.

Figure 4: Packet Viewer



If you want your log records to contain packet data, specify the log packet action in your rules. For details on capturing packets, see “Logging Packets” on page 73. In the getting_started template, the Sensor collects packet data for all events.

Viewing Critical and Major Severity Attacks

If you receive large numbers of daily logs, you might find analyzing log records easier if you focus on one or two attack-severity levels at a time. It is always a good idea to look at critical and major severity attacks first because they can do the most damage to your network.

After you have identified and managed real attacks in the critical and major severity groups, you can analyze log records for the minor and warning severity attacks.

Finding Attacks in Log Records

The easiest way to view log records by severity level is through the Log Viewer component of the UI.

1. Select Log Viewer.
2. In the Severity column, right-click and select Filter > Edit.
3. Select the Critical and Major checkboxes.
4. Click OK.

Using Multiple Rules to Identify an Attack

Use multiple rules for the same host to be notified about specific attacks:

No.	Match			Look For	Action	Notification
	Source	Destination	Terminate Ma...	Attacks		
1	any	Internal Network	☐	Critical	None	Logging Alert Log Packets(20/0)
2	any	Internal Network	☐	Major	None	Logging Alert

The first rule uses the host as the destination, specifies critical severity attack objects, sets an alert, captures 20 packets, and sends an email notification when the attack is detected. (The email log action is not visible in the rule display.) The second rule uses the host as the destination, specifies major severity attack objects, and sets an alert in the log record when the attack is detected. You can create as many rules as you need in the IDP rulebase.

Responding to Real Attacks

After you identify a real attack in your logs, you must decide what to do about it. In most cases, you should begin dropping malicious connections or packets as soon as you see them. However, if you are not quite comfortable dropping traffic yet, you can simply continue to monitor the attack until you have completed the fine-tuning process and are positive that the attack is real, relevant, and dangerous enough to drop.

Currently, all actions in the Action column of your security policy are set to none. To make it easier to monitor and respond to real attacks, you can set an alert notification for all critical and major attacks in your security policy. Then, use the Log Viewer to filter on these attack severities so you can immediately see them. The Sensor still generates log records for lower attack severities, but it does not display them in the filtered view for attacks with alerts.

To configure alerts for critical and major severity attacks, perform the following steps:

1. In your security policy, select a rule that contains critical or major attacks.
2. Right-click a cell in the Notification column and choose Configure.
3. Select Alert and then click OK. If you wish, you can also configure the rule to log packets and send an email message when an attack is identified.
4. Verify and install the edited security policy on your Sensors.
5. Review the logs in the Log Viewer. If IDP detects a critical or major attack, it logs the event.

Identifying Real Attacks

After you add alert actions for critical and major severity attacks to IDP rules, you can investigate the log records those rules create to determine if the attack is targeted, real, and dangerous enough to actively prevent in the future. Because you set alert notifications only on critical and major severity attacks, you can limit your investigation to the critical and major severity views in the Log Viewer.

To investigate log records with alerts, perform the following steps:

1. In the Subcategory column of a log record, double-click on the attack name.
2. In the General tab of the Signature or Protocol Anomaly Editor, examine the attack properties. Use the description to determine if the attack targets your network and if the attack is dangerous. Double-click on the entries in the Attack Versions section to view more details.
3. In the Extended tab, review the known network-security references for this attack. Use the references and detailed information to determine how dangerous the attack is to your network.
4. In any column of a log record, right-click and choose Show > Packet Data. By default, all rules in your getting_started security policy record the packet in the network traffic that matched a rule. You can increase the number of packets captured.
5. Examine the packets used in an attack on your network to help determine the extent of the attempted attack and its purpose, if the attack was successful, and any possible damage to your network.

NOTE: You might need to increase the number of packets that are logged for the attack.

You should always update important software applications regularly, as upgrading to a newer version or patching a vulnerable version with a security fix can make your network harder to compromise. Use the Profiler and Security Explorer to help you locate unpatched systems on your network. For details, refer to the *NetScreen-Security Manager Administrator's Guide.*

Detecting Internal Attacks

You can customize your security policy to detect both *external* and *internal* attacks. For internal attacks, in the Source column of a rule in the Main rulebase, replace any with the address objects that represent your internal hosts. Verify and install the edited security policy on your Sensor, then review your log records in the Log Viewer. For attacks that originate from your address objects, IDP displays the address object name as the source, indicating that the source of the attack was internal.

Preventing Attacks with Multi-Detection Methods

In this step, you explore other rulebases in the IDP system and learn how they use specific detection methods to detect and prevent attacks. So far, you have worked only with the IDP rulebase, but your security policy also contains rules in the following rulebases:

- Traffic Anomalies
- SYN-Protector
- Backdoor Detection

Together, the rulebases provide a Multi-Method Detection (MMD) system that can thwart almost any attack.

Log records generated by all rulebases are combined in the Log Viewer. To see the rule that generated the log, select the All Fields tab to determine which rule number generated the log. Then, right-click the log record and select Goto Policy to open that rulebase.

Log records without a rulebase are generated by a device's Sensor settings, which control how the IDP Sensor handles traffic. You can adjust how these rules are applied by editing the default values for Sensor settings, such as load-time, router, and run-time parameters.

Preventing Traffic Anomaly Attacks

This rulebase protects your network from attacks by using traffic-flow analysis to identify attack patterns over multiple connections. It is designed to detect scans and other distributed network attacks.

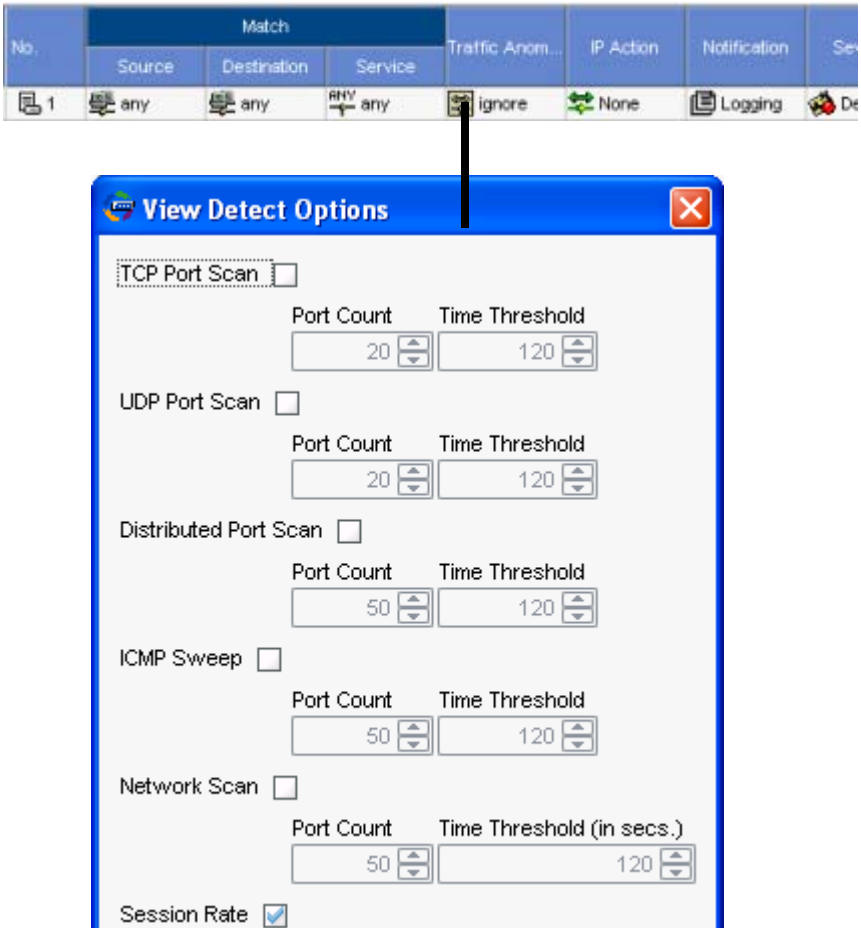
The Traffic Anomalies rulebase looks for patterns that indicate abnormal network activity. Attackers often use scanning tools to automate their port scans, allowing them to scan multiple ports quickly and efficiently. IDP can detect these scans by counting the number of ports scanned in a specified period.

NOTE: When IDP is in-line, you can prevent scans by blocking the attacker's IP address.

To add and modify a Traffic Anomalies rule, perform the following steps:

1. Add a Traffic Anomalies rulebase by clicking the + button in the policy, then selecting Add Traffic Anomalies Rulebase.
2. Click the Traffic Anomalies tab in the Security Policy component to view the Traffic Anomalies rulebase.
3. Click the + button in the Traffic Anomalies rulebase to create the first rule.
4. Right-click in the Traffic Anomalies column to display or change threshold and port parameters, as illustrated in Figure 5.

Figure 5: Editing Traffic Anomaly Parameters



Editing Traffic Anomaly Rules

If you are receiving too many log records for port scans, edit the Traffic Anomalies rulebase to increase the port-count threshold. This increases the number of ports that must be accessed before IDP considers the activity a scan. In the Traffic Anomalies rulebase, perform the following steps:

1. Right-click the Traffic Anomalies column and then choose **Detect** to display the Traffic Anomalies settings.

2. Increase TCP and UDP Port Counts (the number of ports scanned) from 20 to 50.

3. Verify and install the edited security policy on your Sensor.

4. Examine the logs in the Log Viewer.

IDP continues to detect scans on your network but does not generate a log record until 50 ports have been scanned.

About Port and Network Scans

Before attempting to enter an unknown network, attackers gather information about the network and analyze any weaknesses to determine the best attack method. A port scan or network scan is often the first reconnaissance performed. Attackers typically use a scanning tool that attempts to connect to every port on a single machine (port scanning) or connect to multiple IP addresses on a network (network scanning). By determining which services are allowed and responding on your network, attackers can gain valuable information about your network configuration.

Preventing Backdoor Attacks

The Backdoor Detection rulebase protects your network from dangerous backdoors (such as Trojans) by detecting interactive traffic. The rulebase looks at network-traffic patterns and uses heuristics of packet transmissions to detect interactive traffic, a common sign of an attacker using a Trojan or backdoor.

A backdoor is a mechanism installed on a host computer that facilitates unauthorized access to the system. Attackers who have already compromised a system typically install backdoors to make future attacks easier. However, when attackers send and retrieve information to and from the backdoor program (as when typing commands), they generate interactive traffic that IDP can detect.

Unlike antivirus software, which scans for known backdoor files or executables on the host system, IDP detects the interactive traffic that is produced when backdoors are used. This method ensures that IDP can detect all backdoors, both known and unknown, even if the data is encrypted.

NOTE: When IDP is in-line, you can prevent interactive traffic by blocking the attacker's IP address.

To add and modify a Backdoor rule, perform the following steps:

1. Add a Backdoor rulebase by clicking the + button in the policy, then selecting Add Backdoor Rulebase.
2. Click the Backdoor tab in the Security Policy component to view the Backdoor rulebase.
3. Click the + button in the Backdoor rulebase to create the first rule.

The default rule appears, as shown in Figure 6.

Figure 6: Default Rule in Backdoor Detection Rulebase

The rule tells the IDP to detect interactive traffic for all services, but to let it through. You can modify this rule to suit your needs.

Editing Backdoor Detection Rules

Interactive traffic usually indicates human involvement. It looks different from other traffic because humans are manually controlling one end of the connection. In a connection between two programs, the data transfer is automated; TCP packets can be batched and sent in bulk for efficiency. In a connection between a program and a user, packets are sent when they become available; characters display as they are typed (not after the word is complete).

Interactive programs transmit several short IP packets containing individual keystrokes and their echoes, reflecting the real-time actions of a user (or an attacker). Backdoor attacks are usually perpetrated by interactive traffic.

A service you allow on your internal network might produce interactive traffic that matches a Backdoor Detection rule, resulting in false positives and unnecessary log records. To eliminate these log records, you can tell IDP to stop detecting interactive traffic for that service by performing the following steps:

1. In the Log Viewer, investigate your log records for the Backdoor Detection rulebase and identify the services that produce interactive traffic. If you allow the service on your network, the log record might be a false positive. You should ignore the service when detecting interactive traffic.

2. In the Backdoor Detection rulebase, copy the rule, then paste the copy above the existing rule.

3. In the new rule, right-click the Service column and select Select Service to display the Service Object browser. Add the service object that represents the allowed interactive service, then click OK.

4. Right-click the Operation column and select Ignore.

5. Verify the edited security policy, then update the policy on your Sensor.

6. Review the logs in the Log Viewer.

IDP continues to detect interactive traffic for all services not specified in the first rule of the Backdoor Detection rulebase.

Preventing SYN-Flood Attacks

This rulebase protects your network from SYN-flood attacks. This attack attempts to flood your server with TCP requests and overwhelm your resources. IDP detects and prevents SYN-floods by ensuring that the TCP handshake is performed correctly.

In Sniffer mode, the default SYN-flood rule tells IDP to passively monitor the transfer of packets between the client host and the server using a timer to ensure that connections are established promptly. The timer IDP uses for the connection establishment is shorter than the timer the server uses for the connection queue.

If the client host does not send an ACK packet to the server, as would be the case during a SYN-flood attack, the IDP connection timer expires and the default SYN-Protector rule is triggered. IDP creates a log record for the SYN-flood attack.

NOTE: When IDP is in-line, you can actively prevent SYN-floods by dropping connections that are not established promptly.

Click the SYN-Protector tab in the Security Policy component to view the SYN-Protector rulebase. The default SYN-Protector rule is displayed, as shown in Figure 7.

Figure 7: Default SYN-Protector Rule

Editing SYN-Protector Rules

You might not want to detect SYN-floods against specific address objects on your network. To stop detecting SYN-floods against specific hosts, perform the following steps:

1. In the SYN-Protector rulebase, copy the existing rule and paste it as the first rule in the rulebase.

2. In the new rule, right-click the Destination column and select Add to add the address objects you do not want to protect from SYN-floods.

3. In the new rule, right-click the Mode column and select none.

4. Verify and install the edited security policy on your Sensor, then review the logs in the Log Viewer. IDP continues to detect SYN-floods against all address objects not specified in the first rule of the SYN-Protector rulebase.

Chapter 4

Analyzing Your Network

This chapter covers the following topics:

Full instructions for using the Profiler, Security Explorer, and Dashboard can be found in the discussion about analyzing your network in the *NetScreen-Security Manager Administrator's Guide.* This chapter gives an overview of the features.

Application Volume Tracking

Application Volume Tracking (AVT) uses the Profiler to collect fine-grained traffic statistics aggregated over particular time intervals. These statistics can then be viewed using command line utilities or copied off the Sensor to be viewed by third-party reporting applications.

Turning On AVT Collection

To begin collecting data:

1. Make sure the AVT is running on the Sensor.

   scio const -s s0:flow get sc_periodic_stat_update

   you should get a response saying that sc_periodic_stat_update is set to 0x1. If you do not, run this command:

   scio const -s s0:flow set sc_periodic_stat_update 1

2. In NSM, start the Profiler for that Sensor.

   a. Select Devices > IDP Profiler > Start Profiler.

   b. Check the checkbox for the Sensor.

   c. Click OK.

Application Volume Tracking will begin collecting data.

For more information about configuring the Profiler, refer to the *NetScreen-Security Manager Administrator's Guide.*

How AVT Data is Collected and Stored

The Sensor uses a round-robin database to store data in 15-minute and 1-hour intervals. The Sensor stores up to four sets of each interval at a time (four 15-minute intervals and four 1-hour intervals). After it has accumulated four intervals, it begins dropping the oldest interval when it collects a new one.

The Sensor stores the intervals the following directory:

/usr/idp/device/var/stat

There are two subdirectories in `/usr/idp/device/var/stat`:

1hour
15min

Within these subdirectories are several files. One, called `current.stat`, collects the current batch of data. The other files stored previous batches of data that haven't been deleted yet by the round robin database manager.

Use the statview command, described below, to view the stored data. Data is stored in the following table format:

Table 4: AVT Data Format

Source IP	Destination IP	Protocol number*	Source Port	Destination Port	Number of Bytes	Number of Packets

* See http://www.iana.org/assignments/protocol-numbers for protocol-number-to-protocol-name mappings.

Viewing AVT Collections using the CLI

Use the statview command to review collected statistics. You can view meta information about the current tables, view raw data from a particular time interval, run a query for an interval, or chart data for an interval.

statview command options

meta

Displays all stored intervals and the times they cover. Run this command to find out what time intervals are currently stored on the Sensor.

Usage: statview meta

view

Displays the raw data collected for the specified period of time.

Usage: statview view *start_time interval*

where *start_time* is of the format mm:hh:DD:MM:YYYY:(dst|std). It must match one of the interval timestamps available, as shown by running the statview meta command.

where *interval* is 1H or 15M.

You must specify either dst (for local time) or std (for GMT) in your time specification.

Example: statview view 0:16:8:7:2006:dst 1H

This example displays one hour's worth of data starting at 4 PM on July 8, 2006, local time.

query

Displays bytes and packets for an interval collated by one of the following criteria:

- source and destination IP addresses (-a ip)
- protocol (-a proto)
- destination port (-a port).

Usage: statview query [-a *criteria] start_time interval*

where *criteria* is ip, proto, or port, depending on whether you want the data collated by IP address, protocol, or destination port. If you leave out the *-a criteria* portion of the command line, the query command displays all information.

where *start_time* is of the format mm:hh:DD:MM:YYYY:(dst|std). It must match one of the interval timestamps available, as shown by running the statview meta command.

where *interval* is 1H or 15M.

You must specify either dst (for local time) or std (for GMT) in your time specification.

Example: statview query -a port 0:16:8:7:2006:dst 1H

This example displays the total byte and packet counts for each destination port over one hour starting at 4 PM on July 8, 2006, local time.

chart

Displays the total number of bytes and packets for all collections of the specified interval value.

Usage: statview chart *interval*

where *interval* is 1H or 15M.

Example: statview chart 1H

This example lists all the one-hour intervals currently saved, along with their total bytes and packets.

Viewing AVT Data using Other Methods

Interval files are added and deleted in a round robin fashion. You can save older files by copying them to another location before they are deleted. To copy files on a regular basis, create a script and a cron job to copy the files.

You can use the statview command to view these copied files. To run the statview commands on a file other than the default files, use the following command format:

statview -d *db_directory command* -r *filename command_options*

where

- `db_directory` is the directory where you save the files
- `command` is view, query, etc.
- `filename` is the name of the file where you saved the data
- `command_options` is start_time and interval information.

You can also save the output to a text file by redirecting the output. Use the following command template:

statview -d *db_directory command* -r *filename command_options* > *textfile.txt*

About the Dashboard

The Dashboard provides the following at-a-glance displays in one interface:

- Near-realtime monitor of source and destination watch lists
- Top 10 attacks of the previous hour
- Current status of all managed devices

Refer to the *NetScreen-Security Manager Administrator's Guide* for more information.

About the Profiler

The Profiler is a network-analysis tool that helps you learn about your internal network, enabling you to create effective security policies and minimize unnecessary log records. After you configure the Profiler, it automatically learns about your internal network and the elements that comprise it, including hosts, peers (which host is talking to which other host), ports (non-IP protocols, TCP/UDP ports, RPC programs), and layer 7 data that uniquely identifies hosts, operating systems, applications, commands, users, and filenames.

The Profiler is supported in all standalone IDP modes and in HA configurations, and it queries and correlates information from multiple IDP Sensors.

To use the Profiler, you must first configure the networks and hosts on your internal network that you want to monitor. The Sensor monitors traffic at the network and application levels. You can use this data to investigate and analyze potential problems in the network and resolve security incidents.

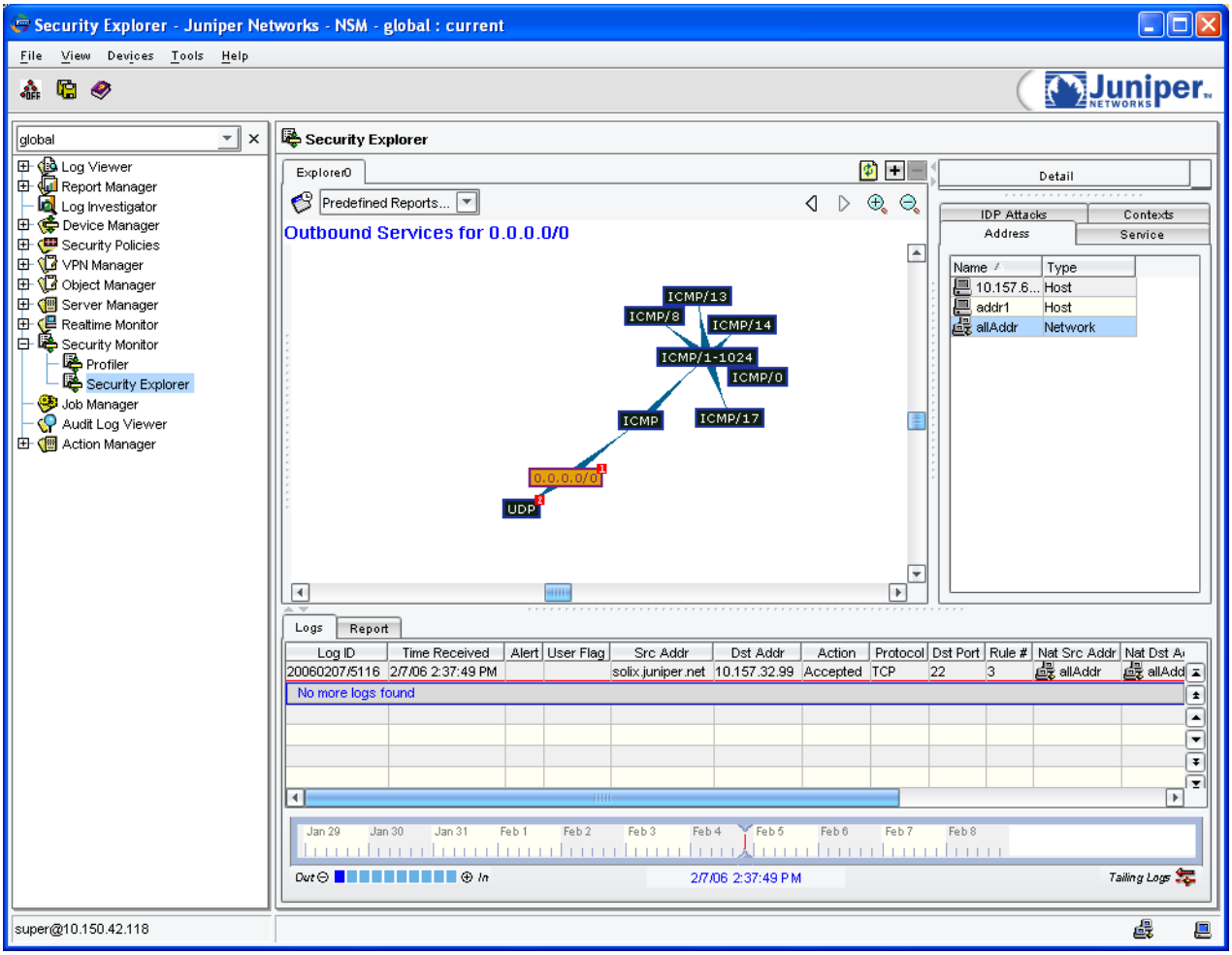
During profiling, the Sensor records network activity at layer 3, layer 4, and layer 7 and stores this information in a searchable database called the Profiler database. The Sensor uses session creation, session teardown, and protocol contexts to generate this database, which defines all unique activities occurring on your network. Unique activities include attempts, probes, and successful connections. The Sensor logs normal events once and all unique events as often as they occur. A normal event is an event that reoccurs frequently and does not change. A unique event is an event that is new or unexpected or that does not match the normal traffic patterns of your network.

Refer to the *NetScreen-Security Manager Administrator's Guide* for more information.

About Security Explorer

NSM Security Explorer is a powerful, graphical tool that enables you to visualize and correlate network behavior based on data collected in the Profiler, Log Viewer and Report Manager. The main component is a graph that represents the relationships between data objects such as hosts, services, attacks, and so on.

Figure 8: Security Explorer



Refer to the *NetScreen-Security Manager Administrator's Guide* for more information.

Chapter 5

Working with Log Records

When directed by the installed security policy, your Sensor generates a log record for security events. In addition, events important for security audits also generate log entries. These log records appear in the Log Viewer component of the NetScreen-Security Manager user interface (NSM UI), where you can analyze them to determine the effectiveness of your current security policy.

Detailed information on viewing and working with IDP logs is contained in the *NetScreen-Security Manager Administrator's Guide* and in the NSM *Online Help*.

Chapter 6
Using Reports

Along with monitoring and logging capabilities in NetScreen-Security Manager (NSM), reports enable you to track and analyze network traffic, activities, and potential attacks. Reports are a standard feature in NSM and are covered in the *NetScreen-Security Manager Administrator's Guide.*

Overview

Reports provide a high-level overview and summary of the log-record data generated by the Sensor(s) deployed in your network. You can use one of the predefined reports such as Top Scan Sources, Top Attackers, or Top Targets to help you track suspicious network activity and identify attack trends. For more specific analysis, you can also design and create custom reports.

Reports provide the following additional benefits:

- Graphical Data Representation
- Integration with Logs
- Central Access to Management Information

Graphical Data Representation

You can use reports to view log data in both tabular and graphical form. The various depictions of the data make it easier to identify trends and potential areas of risk. Depending on your preference, you can also choose to view the data in either a horizontal bar graph or a pie chart.

Integration with Logs

Reports are also integrated with the Log Viewer and Log Investigator modules. By simply clicking on a data point depicted in a report, you can quickly drill down to access and view the specific log entries presented in the report data.

Central Access to Management Information

For network administrators and security analysts interested in tracking and identifying potential network trends and attacks, reports provide a single, graphical view into the network.

Topics Covered in the NSM Administrator's Guide

The discussion about reporting in the *NetScreen-Security Manager Administrator's Guide* covers the following topics:

- Report Types. (IDP reports are in the DI/IDP grouping.)
- Custom and Predefined Reports
- Working with Reports
- Generating Reports
- Generating Reports on a Schedule
- Exporting Reports

See the *NetScreen-Security Manager Administrator's Guide* for more information about these topics.

Chapter 7

Creating Effective Security Policies

This chapter covers the following topics:

- Understanding Detection Methods
- Understanding Rules and Rulebases
- Designing Rules
- Working with Rulebases
- Managing Security Policies

Security policies define what traffic to look at, what to look for, and how to respond when an attack is detected. To create and manage security policies, use the Security Policy Editor, a user interface (UI) component.

Security policies consist of detection methods, rules, and rulebases.

For more information about managing IDP security policies through NetScreen-Security Manager (NSM), refer to the *NetScreen-Security Manager Administrator's Guide.*

Understanding Detection Methods

The IDP system uses multiple detection methods to identify and prevent attacks. By combining these detection methods and using them simultaneously, IDP accurately and efficiently detects threats to your network. The IDP Multi-Method Detection (MMD) mechanism integrates the following detection mechanisms:

- Stateful signatures
- Protocol anomalies
- Backdoor detection
- Traffic anomalies
- IP spoofing
- Layer 2 Attacks

- Denial-of-Service (DoS) detection
- Network honeypot

The following sections detail each detection method.

Stateful Signatures (IDP Rulebase)

The IDP system uses *stateful* signatures to detect known attacks. A stateful signature is a signature that not only knows the pattern it is attempting to find, but also knows where to look for that pattern.

Stateful signatures produce very few false positives because they understand the context of the attack and can eliminate huge sections of network traffic they know the attack would not be in.

Stateful signatures are much smarter than regular signatures that are used by other intrusion detection systems: Stateful signatures know the protocol or service used to perpetrate the attack, they know the direction and flow of the attack, and they know the context in which the attack occurs.

Obviously, though, a signature cannot contain all this information within the attack signature pattern—the data must be associated with the signature, but not actually part of the pattern itself.

IDP does this by combining the attack pattern with service, context, and other information into a signature attack object.

NSM includes hundreds of signature attack objects that all use stateful signatures to detect known attacks. You can view, create, edit, or delete signature attack objects using the Signature Editor in the NSM UI.

NOTE: You can create your own signatures to use in rules. For more information, see “Managing Attack Objects” on page 93.

Protocol Anomalies (IDP Rulebase)

Protocol anomalies are deviations from the protocol standard. Most legitimate traffic adheres to the published specifications (RFC) for protocols, but traffic that does not produces an anomaly. Illegal or ambiguous traffic does not just happen—often an attacker creates an anomaly for a specific purpose, like evading an intrusion detection system (IDS).

The IDP system uses the RFC to create protocol-anomaly attack objects for deviations from the protocol’s published specification. These protocol-anomaly attack objects are then used to detect protocol anomalies in your network traffic. The IDP system includes all known protocol anomalies as attack objects to detect unknown attacks.

Backdoor Detection (Backdoor Detection Rulebase)

Backdoor detection uses network traffic patterns and heuristics of packet transmissions to detect interactive traffic, a common sign of an attacker using a Trojan or backdoor.

Traffic Anomalies (Traffic Anomalies Rulebase)

A traffic anomaly is a pattern that indicates abnormal network activity. Attackers create traffic anomalies when they use a scanning tool (which automates port scans) to map your network. The IDP system counts the number of ports scanned in a specified time period and uses this traffic flow analysis to identify scans. Traffic flow analysis can also identify other attacks that occur over multiple connections and sessions.

IP Spoofing (Sensor Settings in Device Manager)

IP spoofing occurs when a packet uses a fake IP address. Attackers, who typically do not want you to know where an attack is coming from, often use a fake IP address to disguise the real source address of a packet. The IDP system detects IP spoofing by comparing the IP addresses of packets to the IP addresses of devices on your network. An IP address is considered spoofed if:

- An incoming packet uses an IP address that belongs to a network object on your internal network.
- An outgoing packet uses an IP address that does not belong to a network object on your internal network.

You can set IP spoofing detection for each IDP Sensor by editing the Sensor's settings in Device Manager. Click Anti-Spoof Settings. Click the + button, then specify a Sensor interface and specify the network objects that are attached to that interface.

Layer 2 Attacks (Sensor Settings in Device Manager)

Layer 2 is the Network Layer in the TCP/IP Model. Layer 2 protocols manage data transfer from the source address to the destination address. Attackers can manipulate Layer 2 protocols to perform ARP attacks (such as ARP cache poisoning) and other MAC attacks.

The IDP system detects Layer 2 attacks by defining implied rules on the IDP Sensor. Implied rules include ARP table restrictions, fragment handling, connection timeouts, byte/length thresholds for data packets, and other mechanisms. You can adjust the settings for these implied rules in the Sensor settings in Device Manager, described in the NetScreen-Security Manager online help under Sensor Settings.

Denial-of-Service Detection (SYN-Protector Rulebase)

A Denial-of-Service (DoS) occurs when too many connection requests overwhelm and exhaust all the allocated resources on a system. A SYN-flood, in which an attacker manipulates the basic TCP three-way handshake to overflow a system's connection table, is a common DoS attack. The IDP system uses DoS detection to prevent SYN-floods from occurring on your network.

Network Honeypot (Network Honeypot Rulebase)

The IDP network honeypot impersonates open ports on existing servers. Attackers typically perform port scans or other information-gathering activities on your network to help them create a targeted attack designed to compromise your system. The IDP system uses fake ports to detect these reconnaissance activities.

NOTE: The Network Honeypot rulebase cannot impersonate ports on the physical IDP Sensor. In other words, if the Sensor has an IP address of 1.1.1.1, it cannot impersonate 1.1.1.1:80.

Understanding Rules and Rulebases

Rules give context to detection mechanisms by specifying which part of the network traffic the IDP system should look in to find attacks. When a rule is matched, an attack has been detected in the network traffic for that rule.

Multiple rules go into a rulebase. One or more rulebases go into a security policy. A security policy, once it is pushed to a Sensor, governs how that Sensor responds to traffic by setting an action. When a rule match triggers the action for that rule, the IDP system performs the specified action and protects your network from that attack.

You create rules using the Security Policy Editor in NSM. Each rulebase can have multiple rules—you determine the sequence in which rules are applied to network traffic by placing them in the desired order.

Each rulebase in the IDP system uses specific detection methods to identify and prevent attacks. Together, rulebases provide an MMD system that can thwart almost any attack.

You can use NSM to add, modify, disable, and delete rules in six rulebases:

- IDP
- SYN-Protector
- Backdoor Detection
- Network Honeypot
- Exempt
- Traffic Anomalies

For more information about the different IDP rulebases, see “Working with Rulebases” on page 75 and the *NetScreen-Security Manager Administrator’s Guide* and the NSM *Online Help*.

A security policy is the combination of all rulebases (and rules) into a comprehensive plan that defines how the IDP system works on your network.

You install your security policy on your IDP Sensors using NSM. For more information about managing security policies, see “Managing Security Policies” on page 88.

Designing Rules

Well-designed rules can make detection and prevention of attacks easy and manageable. Conversely, poorly designed rules can make detection and prevention difficult and frustrating—or worse, leave your network vulnerable.

This section covers the basics of designing rules for your network and includes the following guidelines:

- Identifying your network risks and threats
- Using attack objects to detect attacks against your network vulnerabilities (IDP rulebase only)
- Setting IDP actions that protect your network
- Understanding how terminate match and non-terminal rules work in the IDP system
- Dealing with false positives

Although each IDP rulebase uses different detection mechanisms to detect attacks, they all share similar rule-design concepts. After you have learned the basic concepts of designing an efficient rule, you can learn more about building rules for each IDP rulebase in “Working with Rulebases” on page 75.

Identifying Risks and Threats

To identify risks and threats to your network, you must first understand your network and the type of traffic that occurs on it. Once you know what you are vulnerable to, it becomes much easier to create a strategic defense against attackers. Your first steps in securing your network are as follows:

- Understand your existing network topology. This means identifying which services are allowed on your network, which ports are open, and where important files are stored.
- Know your normal traffic. This means using your knowledge of your network to clearly define what should happen—and what should not happen on it.

Remember, you know your network; attackers do not. This gives you a tremendous advantage when protecting your network, so make use of it. You should start by determining which devices on your network might be attractive to attackers. Using the Profiler is an ideal way to determine your network vulnerabilities; for details, see “Analyzing Your Network” on page 43.

Setting Source and Destination

All attacks have a source (where the attack is coming from) and a destination (where the attack is going to). You might not always know both the source and destination of an attack, but you probably do know one of them. Typically, a server or other network object on your network is the destination for incoming attacks and can sometimes be the source for interactive attacks (see “Using the Backdoor Detection Rulebase” on page 87 for more information about interactive attacks).

In the IDP system, network objects represent the devices running on your network. You use the Object Editor in the NSM UI to create a network object for each device, then use these network objects in rules to specify the source and destination of attacks. See the *Online Help* topic “Adding Network Objects” for more information.

Example: Detecting Incoming Attacks

You want to detect incoming attacks that target your internal network. Set the source to any and select the network object that represents the host or server you want to protect from attacks as the destination. Your rule looks similar to this example:

No.	Match		
	Source	Destination	Service
1	any	any	any

Example: Detecting Attacks Between Networks

You want to detect attacks between two networks. Select multiple network objects as the source and destination. Your rule looks similar to this example:

No.		
	Source	Destination
1	Europe Mail Server Europe Workstation Europe Users	Security Auditing Group Security Team Network Web Server Group

The more specific you are in defining the source and destination of an attack, the more you reduce false positives.

Setting Services

Services are Application Layer protocols that define how data is structured as it travels across the network. Because the services you support on your network are the same services that attackers must use to attack your network, you can specify which services are supported by the destination to make your rule more efficient.

In the IDP system, service objects represent the services running on your network. IDP comes with several service objects based on industry-standard services already created for you. You use these service objects in rules to specify the service an attack uses to access your network.

Because the HTTP service can use multiple common or nonstandard ports, IDP register HTTP on the standard TCP port, TCP/80, as well as the following ports:

- 7001 (Weblogic)
- 8000, 8001, 8080 (common HTTP ports)
- 8100, 8200 (JRun)
- 8888 (Oracle 9i)
- 9080 (Websphere)

When the Sensor detects HTTP traffic on these ports, IDP automatically analyzes the HTTP contexts to detect possible protocol anomalies, and signature attack objects that use HTTP as the default service automatically recognize the traffic as HTTP. When the Sensor detects non-HTTP traffic on these ports, IDP does not analyze HTTP context or check for HTTP protocol anomalies.

Using Default Services

Attack objects have default services associated with them. When you select an attack object in the Attack column, the service associated with that attack object becomes the default service for the rule. To see the exact service, view the attack object details.

NOTE: The Service column does not appear in Compact mode. To see it, select View > Show Expanded Mode.

Example: Default Service

You want to protect your webserver from HTTP attacks. Set the service to Default, and add an attack object that detects critical HTTP attacks. The Service column in the rule still displays Default, but the rule actually uses the default service of HTTP, which is specified in the attack object.

Your rule looks similar to this example:

Destination	Service	Terminate ...	Attacks
Web Server	Default	☐	HTTP - Critical

Using Custom Services (Service Objects)

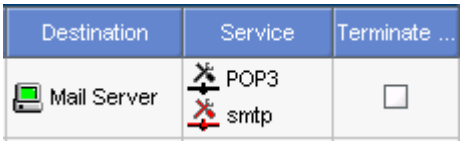
If you do not want to use the default service(s) of the attack objects, you can select specific services.

NOTE: The Service column does not appear in Compact mode. To see it, select View > Show Expanded Mode.

Example: Service Objects

Your mail server supports POP3 and SMTP connections, but not IMAP. Set TCP-POP3 and TCP-SMTP service objects as services that can be used to attack that server. Because IMAP is not supported, you do not need to add the TCP-IMAP service object.

Your rule looks similar to this example:

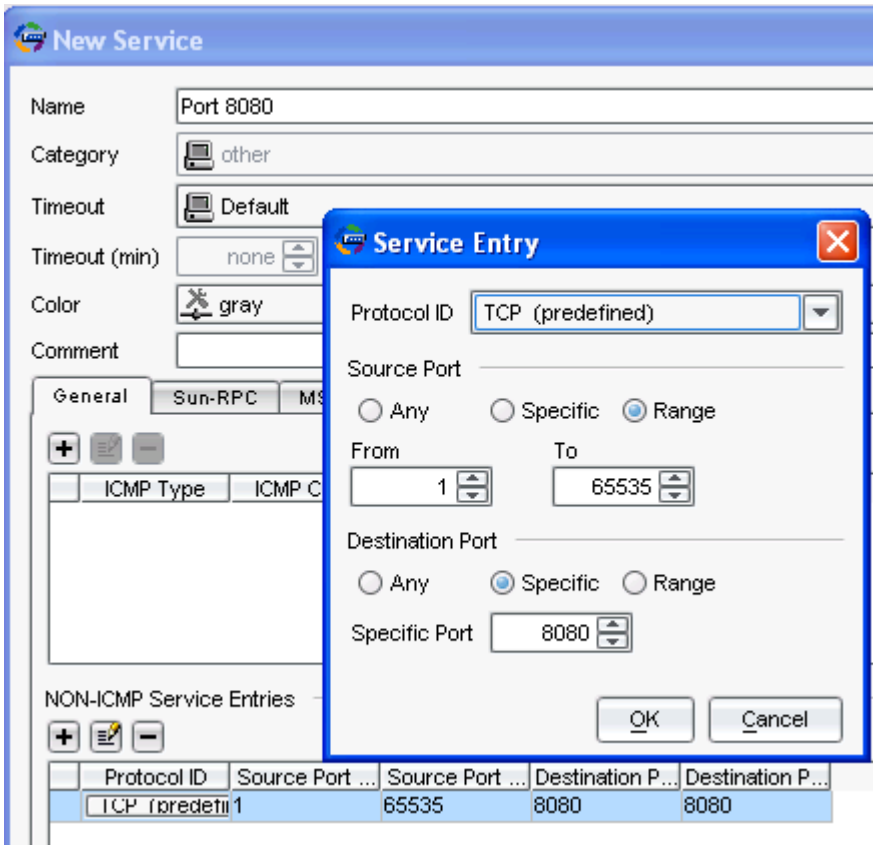


If you are supporting services on nonstandard ports, you should choose a service other than default.

Example: Nonstandard Services

You use a nonstandard port (8080) for your HTTP services. Use the Service Editor to create a service object on port 8080. Add this service object to your rule, then add several HTTP attack objects, which have a default service of TCP/80. IDP uses the specified service, HTTP-8080, instead of the default and looks for matches to the HTTP attacks in TCP traffic on port 8080.

Figure 9: Service Editor



Your rule looks similar to this example:

Destination	Service	Terminate ...
Web Server	Port 8080	☐

You can create your own service objects to use in rules, such as service objects for protocols that use nonstandard ports. However, you cannot match attack objects to protocols they do not use. Refer to the NSM *Online Help* for more information.

Setting Terminate Match Rules

A terminate match rule ends actions for a connection by terminating the IDP rule-matching algorithm. You can use a terminate match rule for the following actions:

- Set different actions for different attacks for the same source and destination.
- Disregard traffic that originates from a known trusted source. Typically the action is None for this type of terminate match rule.

- Disregard traffic sent to a server that is only vulnerable to a specific set of attacks. Typically, the action is Drop Connection for this type of terminate match rule.

The normal IDP rule-matching algorithm starts from the top of the rulebase and matches all rules with the same source, destination, and service until it encounters the end of the rulebase. A terminate match rule is an exception to this normal rule-matching algorithm: When a match is discovered in a terminate match rule, the Sensor does not continue to apply subsequent rules to that connection, regardless of whether the traffic matches the attack objects in the matching rule.

Use caution when defining terminate match rules. Remember that traffic matching the source, destination, and service of a terminate match rule is not compared to subsequent rules, even if the traffic does not match an attack object in the terminate match rule. (The match only considers source, destination, and service. It does not consider attack objects.) Only use a terminate match rule when you want to examine a certain type of traffic for one specific set of attack objects and no others. You can inadvertently leave your network open to attacks by creating an inappropriate terminate match rule.

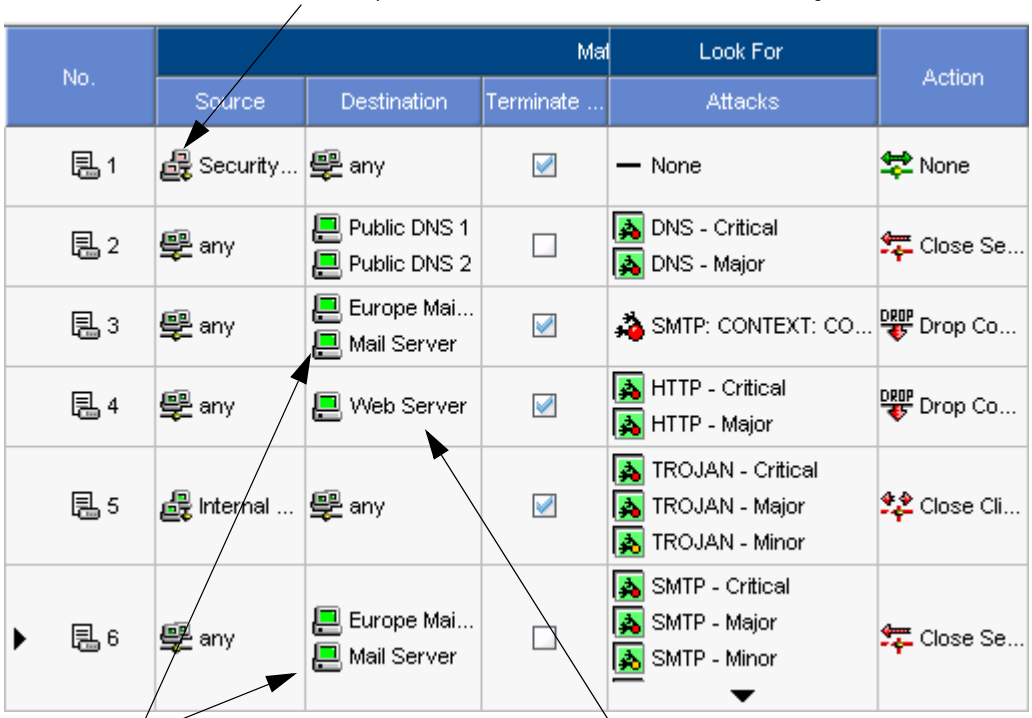
Be particularly careful about terminate match rules using any for both the source and destination.

Terminate match rules should appear *after* any rules that you *do* want processed if the event matches both, but *before* any rules that you *do not* want processed if the event matches both. You set a rule as terminate match by selecting the box in the Terminate Match column of the Security Policy Editor when you create or modify the rule.

Example: Terminate Match Rules

Figure 10: Terminate Match Rules Example

Rule 1 terminates the match algorithm if the source of the traffic originates from the Security Team Network, a known trusted network. If this rule is matched, the Sensor drops the connection and does not continue monitoring the session for malicious data.



Rule 3 terminates the match algorithm when the destination is the Corporate or Europe Mail Server and the attack is an email that uses the SMTP context Confidential.

Rule 4 terminates the match algorithm when the destination is the Web Server group and the attack is from the Critical or High HTTP attacks. The rule ensures that the Sensor drops the most important HTTP attacks against the Web Server, but does not continue to match the connection against other targets.

Setting Attack Objects

Attack objects represent specific patterns of malicious activity within a connection and are a method for detecting attacks. Each attack object detects a known or unknown attack that can be used to compromise your network. From more information about attack objects, see “Managing Attack Objects” on page 93.

NOTE: Attack objects are used to create rules in the IDP rulebase only. Other rulebases use different detection methods.

Attack objects are organized in a hierarchy according to the following:

- First Level: Severities. The severity of the attack, based on lethality and chance of success.
- Second Level: Protocols. The protocols that attacks use to enter your network.
- Third Level: Attack Type. The type of attack, based on how the attack functions (buffer overflow attempt, password exploit, Denial-of-Service, and so on)

You can use the attack objects hierarchy to add attack objects to your rule in groups or individually.

Adding Attack Objects by Severity

NSM groups attack objects by severity to help you choose the attack objects that are the most dangerous to your network. Severities are the first-level groups in the attack-objects hierarchy.

NSM defines five severity groups, each with a recommended set of IDP actions. You can add a severity group to the Attack column in your rule, then choose the recommended actions for the severity group in the Action column. (For more information about actions, see “Setting Actions” on page 67.)

NOTE: To protect a critical network object or popular attacker target (such as your mailserver), use multiple severity groups to ensure maximum protection.

The severity groups are described below:

- Severity 1 (Critical). Critical attacks attempt to evade an intrusion detection system (IDS), crash a machine, or gain system-level privileges. A critical host or appliance can be shut down or overwhelmed with traffic, as in a Denial-of-Service (DoS) attack. A reboot of the host or appliance or more drastic action may be required to correct the damage. Critical attacks can be arbitrary code that could be executed as root or system, or worms that cause damage or install Trojans. *Recommended Action:* Drop, alert, and log.

- Severity 2 (Major). Major attacks attempt to crash or overwhelm a single service while other services on the same host are unaffected. Examples include attacks that can cause a severe (but not necessarily total) DoS to a service, arbitrary code that could be executed as a nonprivileged user, leak of valuable information (account names and passwords, for example), and worms that cause DoS but do not leave backdoors. *Recommended Action:* Drop, alert, and log.

- Severity 3 (Minor). Minor attacks attempt to obtain critical information through directory traversal or information leaks. Examples include attacks that cause moderate DoS, leaks of noncritical information (such as path names), and annoying but harmless worms. *Recommended Action:* Log.

- Severity 4 (Warning). Warning-level attacks attempt to obtain noncritical information or scan the network with a scanning tool. Examples include detection of a scanner, file transfer through instant-message software, and login failures. They can also be obsolete attacks or anomalous (but probably harmless) traffic or protocol violations. Recommended Action: Log.

- Severity 5 (Informational). Informational attacks are normal, harmless traffic containing URLs, DNS lookup failures, and SNMP public community strings. You can use informational attack objects to obtain information about your network, or to monitor for instant messaging, FTP logins, or HTTP access. *Recommended Action:* None.

If you do not want to apply attack objects by severity or do not want to choose an entire severity group for a rule, you can select your attack objects by service.

Adding Attack Objects by Service

Each severity group is subdivided into the services that attacks can use to enter your network. Services are the second-level groups in the attack objects hierarchy. Services are Application Layer protocols that define how data is structured as it travels across the network. A protocol is a specification that indicates how communication between two entities (applications, servers, Ethernet cards, and so on) occurs.

When attacking a system, attackers use the protocol of a supported service to communicate their malicious activity to the server. However, attackers can only use protocols that are supported by the system they are attacking. You can add a service group to the Attack column in your rule; however, you need to select only the services that are used by the network objects you are protecting with the rule.

Example: Attack Objects by Service

You rely on FTP and HTTP for extensive file transfer on your webserver. Choose the FTP and HTTP service groups to carefully monitor all traffic that uses these services.

Example: Attack Objects by Service

Your webserver does not allow SSH, RLOGIN, or RSH access. Do not add the SSH, RLOGIN, or RSH service groups to your rule.

If you do not want to apply attack objects by service or choose an entire service group for a rule, you can select your attack objects by attack type.

Adding Attack Objects by Attack Type

Each service group is subdivided into the attack type (buffer overflow attempts, password exploits, Denial-of-Service, and so on). Attack types are the third-level groups in the attack-objects hierarchy.

You can add an attack type group to the Attack column in your rule. If you do not want to apply attack objects by type or choose an entire group for a rule, you can select your attack objects individually.

Adding Attack Objects Individually

IDP uses three types of attack objects to detect malicious activity on the network: signature, protocol-anomaly, and compound attack objects.

- Signature attack objects detect attacks and intrusion attempts by matching stateful signatures to your network traffic. A signature is a known pattern that exists within an attack; the pattern is selected by analyzing the attack and selecting a key piece of information that, when present, identifies the attack (a segment of code, a URL, a value in a packet header, and so on). A *stateful* signature is a signature that not only knows the pattern it is attempting to find, but also knows where to look for that pattern. IDP uses stateful signatures to reduce false positives and increase detection accuracy for known attacks.

- Protocol-anomaly attack objects detect abnormal or ambiguous messages within a connection according to a set of default rules. Attackers can create protocol anomalies to evade signature-based detection mechanisms. The IDP system uses the protocol RFC to create protocol-anomaly attack objects for each possible deviation from the protocol's published specification.

NOTE: Some software applications produce legitimate traffic that can appear as protocol anomalies due to nonstandard implementation of a protocol RFC.

- Compound attack objects combine multiple signatures and/or protocol anomalies into a single attack object. A compound attack specifies that traffic match multiple patterns or anomalies before triggering the actions defined for a rule, reducing false positives and improving accuracy.

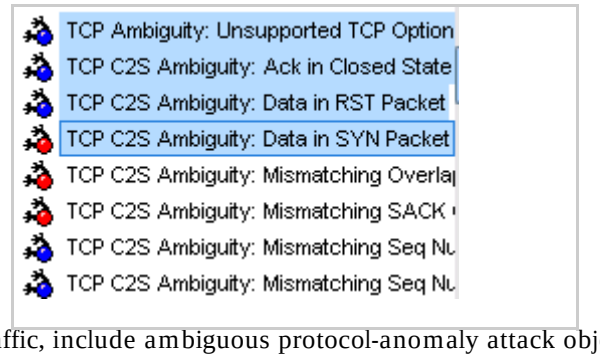
Normalizing Network Traffic (Protocol Normalization)

Some Internet protocols, specifically TCP, are specified by requests for comments (RFCs) that use terminology such as *may implement* and *should implement*. This indeterminate language allows application programmers to write programs that implement the protocol differently.

Attackers can exploit these implementation differences to bypass a firewall or an intrusion detection system (IDS) by sending malicious packets that use ambiguous protocol rules. The firewall or IDS might process the packet differently than the receiving host and possibly allow an attack to reach the protected network.

To prevent this, every sequence of packets that IDP might treat differently than the receiving host should be normalized. IDP can detect and drop any packet that contains ambiguous protocol rules, allowing only messages that conform to the unambiguous rules of the protocol RFC to reach the protected network.

Figure 11: Selecting Ambiguous Protocol-Anomaly Attack Objects



To normalize traffic, include ambiguous protocol-anomaly attack objects in your security policy.

Setting Actions

Network security is an ongoing process of understanding what is normal traffic for your network. Eliminating malicious traffic is important, but identifying ambiguous traffic can be equally important. You do not always want to drop traffic that appears abnormal; you might want to reset the connection, block the attacker, log the event, or take all three actions.

You can tell IDP which actions to perform against attacks that match rules in your security policy. You can set different actions for each rule in your security policy. The IDP system uses two types of actions to protect your network: actions and IP actions.

For each attack that matches a rule:

- Actions respond to matching traffic by ignoring, dropping, or closing the current attacking packets or connection. In some rulebases, actions are known as *operations* (Network Honeypot rulebase) or *modes* (SYN-Protector rulebase).

- IP actions respond to future traffic based on the previous matching traffic by logging, blocking, or dropping future attacking connections. IP actions are a dynamic, specific method for protecting your network from future intrusions while permitting legitimate traffic.

Using Actions Against Current Connections

In the security policy templates, the actions are already chosen for you based on the severity of the attack object. When building your own security policy (or editing an existing policy), you can choose actions that fit the security needs of your network.

If the rule is triggered, IDP can perform actions against the connection. Remember that IDP can drop traffic only when it is in-line; IDP Sensors running in Sniffer mode cannot perform drop or close actions. Each rulebase contains different actions.

The Traffic Anomalies rulebase uses IP actions. For details on IP actions, see "Using IP Actions Against Existing Connections" on page 70. For details about the Traffic Anomalies rulebase, see "Using the Traffic Anomalies Rulebase" on page 76.

NOTE: If a rule that contains an action of None is matched, the corresponding log record displays accept in the action column of the Log Viewer.

If a packet triggers multiple rule actions, the Sensor will apply the most severe action. For example, if the rules dictate that a packet will receive a diffserv marking and be dropped, then the packet will be dropped.

Table 5: IDP Rulebase Actions, Sorted by Severity

Action	Description
None	IDP takes no action against the connection.
Ignore	IDP ignores the remainder of a connection after an attack object in a Security Policy rulebase is matched.
Diffserv Marking	Assigns the service-differentiation value indicated to the packet, then passes it on normally. Set the service-differentiation value in the dialog that appears when you select this action in the rulebase. The marking has no effect in Sniffer mode.
Drop Packet	IDP drops a matching packet before it can reach its destination but does not close the connection. Use this action to drop packets for attacks in traffic that is prone to spoofing, such as UDP traffic. Dropping a connection for such traffic could result in a Denial-of-Service that prevents you from receiving traffic from a legitimate source address.
Drop Connection	IDP drops the connection without sending an RST packet to the sender, preventing the traffic from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.
Close Client and Server	IDP closes the connection and sends an RST packet to both the client and the server. If the IDP Sensor is in Sniffer mode, IDP sends an RST packet to both the client and server but does not close the connection.
Close Client	IDP closes the connection to the client but not to the server.
Close Server	IDP closes the connection to the server but not to the client.

Table 6 shows the Backdoor Detection rulebase actions.

Table 6: Backdoor Detection Rulebase Actions

Action	Description
Accept	IDP accepts the interactive traffic.
Drop Connection	IDP drops the interactive connection without sending an RST packet to the sender, preventing the traffic from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.
Close Client and Server	IDP closes the interactive connection and sends an RST packet to both the client and the server. If the IDP Sensor is in Sniffer mode, IDP sends an RST packet to both the client and server but does NOT close the connection.
Close Client	IDP closes the interactive connection to the client but not to the server.
Close Server	IDP closes the interactive connection to the server but not to the client.

Table 7 shows the Network Honeypot rulebase operations.

Table 7: Network Honeypot Rulebase Operations

Operations	Description
Ignore	IDP does not impersonate the selected destination and service.
Impersonate	IDP creates a counterfeit port based on the selected destination and service.

Table 8 shows the SYN-Protector rulebase modes.

Table 8: SYN-Protector Rulebase Modes

Modes	Description
None	IDP takes no action and does not involve itself in the three-way handshake.
Relay	IDP acts as the middleman, or relay, for the connection establishment, performing the three-way handshake with the client host on behalf of the server. As of IDP 4.0, the Sensor use SYN Cookies.
Passive	IDP handles the transfer of packets between the client host and the server but does not actively prevent the connection from being established. Instead, IDP uses a timer to ensure that connections are established promptly, minimizing the use of server resources. The timer IDP uses for the connection establishment is shorter than the timer the server uses for the connection queue.

Using IP Actions Against Existing Connections

If the current network traffic matches a rule, IDP can perform an IP action against future network traffic that uses the same IP address. The Traffic Anomalies rulebase uses IP actions to handle sudden changes in your network traffic, such as scans and session-rate increases. For details, see “Using the Traffic Anomalies Rulebase” on page 76.

IP actions are similar to other actions; they tell IDP to drop or close the connection. However, because you now also have the attacker’s IP address, you can choose to block the attacker for a specified period. If attackers cannot immediately regain a connection to your network, they might try to attack easier targets.

Use IP actions in conjunction with actions and logging to secure your network. In a rule, first configure an action to detect and prevent current malicious connections from reaching your network objects. Then, right-click in the IP Action column of the rule and select Configure to bring up the Configure IP Action dialog box; configure an IP action to prevent future malicious connections from the attacker's IP address.

Choosing an IP Action

For each IP action option, the IDP system generates an IP action. The IP action instructs the IDP Sensor to perform the specified task. Select from the following options:

- IDP Notify. IDP does not take any action against future traffic but logs the event.
- IDP Block. IDP blocks the matching connection and future connections that match the criteria set in the Blocking Options box.
- IDP Close. IDP closes future connections that match the criteria in the Blocking Options box.

Choosing a Blocking Option

Each blocking option follows the criteria you set in the Actions box. Blocking options are based on the source, destination, or service of the attack traffic:

- Source Subnet, Destination, Dest. Port and Protocol. IDP blocks future traffic based on attack traffic source subnetwork, destination, destination port, and protocol.
- Source, Destination Subnet, Dest. Port and Protocol. IDP blocks future traffic based on attack traffic source, destination subnetwork, destination port, and protocol.
- Source, Protocol. IDP blocks future traffic based on attack traffic source and protocol.
- Source Subnet and Protocol. IDP blocks future traffic based on attack traffic source subnetwork and protocol.
- Source Subnet. IDP blocks future traffic based on attack traffic source subnetwork.
- Destination, Dest. Port and Protocol. IDP blocks future traffic based on attack traffic destination, destination port, and protocol.
- Destination and Protocol. IDP blocks future traffic based on attack traffic destination and protocol.
- Destination Subnet, Dest. Port and Protocol. IDP blocks future traffic based on attack traffic destination subnetwork, destination port, and protocol.
- Destination Subnet and Protocol. IDP blocks future traffic based on attack traffic destination subnetwork and protocol.

- Destination Subnet. IDP blocks future traffic based on attack traffic destination subnetwork.
- Source, Destination, Destination Port and Protocol. IDP blocks future traffic based on attack traffic source, destination, destination port, and protocol.
- Source. IDP blocks future traffic based on attack traffic source.
- Destination. IDP blocks future traffic based on attack traffic destination.
- From Zone, Destination, Destination Port and Protocol. Blocks future traffic from a particular zone to a destination and destination port via a particular protocol. Only useful for ISG Family Firewall/IDP devices. Not useful for standalone IDP Sensors.
- From Zone. Blocks future traffic from a particular zone. Only useful for ISG Family Firewall/IDP devices. Not useful for standalone IDP Sensors.

Configuring IP Logging, Alerts, and Timeouts

When IDP detects an attack traffic match for a rule and triggers an IP action, IDP can log information about the IP action that was taken and/or create an alert flag in the Log Viewer. The Timeout is the number of seconds that you want the IP action to remain in effect after a traffic match. For permanent IP actions, leave the Timeout blank.

NOTE: The security policy templates have a default IP action of none.

Setting Notification

The first time you design a security policy, you might be tempted to log all data for all attacks and let the policy run indefinitely. Do not do this! Some attack objects are informational only, and others can generate false positives and redundant logs. If you become overloaded with data, you can miss something important. Remember that security policies that generate too many log records are hazardous to the security of your network, as you might discover an attack too late or miss a security breach entirely as a result of having to sift through hundreds of log records. Excessive logging can also affect IDP throughput, performance, and available disk space. A good security policy generates enough logs to fully document only the important security events on your network.

To log an attack for a rule, right-click the Notification column of the rule and select Configure. The Configure Notification dialog box appears. Select Logging, then click OK. Each time the rule is matched, the IDP system creates a log record that appears in the Log Viewer.

In addition, you can specify that the log entry be marked as an Alert, and you can capture packets from around the event. Use the Logging tab to set these preferences.

You can choose to simply log an attack and create log records with attack information that you can view real-time in the Log Viewer. For more critical attacks, however, you might want to be notified immediately by email, have IDP run a script in response to the attack, or set an alert flag to appear in the log record. Your goal is to fine-tune the attack notifications in your security policy to your individual security needs.

NOTE: For details about fine-tuning, see "Fine-Tuning Security Policies" on page 23.

Depending on your security needs and the severity of the attack, you might want IDP to provide additional notification when a rule is matched. Some additional notification features use specific log parameters, as shown in Table 9.

Table 9: Log Parameters

Parameter	Value
< action >	Action taken by the IDP Sensor, as defined in the matching rule.
< attack >	[For attacks only] Name of the attack object that triggered the rule. Use the attack, category, and subcategory log record fields to determine the nature of the attack.
< bytes >	Number of bytes in connection, if accounting turned on.
< category >	Information about how log was categorized by Sensor.
< comment >	User-specified value.
< day id >	GMT-based day log was generated on (for example: yyyymmdd).
< dst addr >	Packet's destination address (for example: A.B.C.D).
< dst port >	Packet's destination port number (for example: 0-65535).
< elapsed >	Number of seconds in connection, if accounting turned on.
< in nic >	Name of the inbound NIC.
< is alert >	Did the policy specify an alert (for example: yes/no)?
< is duplicate >	Is this a duplicate log (for example: yes/no)?
< is hidden >	Has the user deleted this log (e.g. yes/no)?
< misc >	Miscellaneous data set by Sensor.
< nat dst addr >	Packet's NAT destination address (for example: A.B.C.D).
< nat src addr >	Packet's NAT source address (for example: A.B.C.D).
< nat dst port >	Packet's NAT destination port number (for example: 0-65535).
< nat src port >	Packet's NAT source port number (for example: 0-65535).
< out nic >	Name of the outbound NIC.
< packets >	Name of packets in connection, if accounting turned on.
< policy name >	Name of the policy that triggered the log.
< policy ver >	Version of the policy that triggered the log.
< protocol >	Protocol the connection was using (TCP, UDP, ICMP, and so on).
< record id >	(n-1)th log in day.
< rulebase >	Name of the rulebase that triggered the log.
< rule number >	Name of the rule number that triggered the log.
< run script >	Did the policy request a script to be run (e.g. yes/no)?

Parameter	Value
< send email >	Did the policy request e-mail msg to be sent (e.g. yes/no)?
< send snmp >	Did the policy request an snmp msg to be sent (e.g. yes/no)?
< send syslog >	Did the policy request a syslog msg to be sent (e.g yes/no)?
< sensor addr >	Sensor's address (e.g. A.B.C.D).
< sensor vin >	Sensor's VIN.
< session id1 >	Internal tracking number.
< session id2 >	Internal tracking number.
< severity >	Severity specified by the policy.
< src addr >	Packet's source address (for example: A.B.C.D).
< src port >	Packet's source port number (for example: 0-65535).
< subcategory >	Detailed information about how log was categorized by Sensor.
< timestamp >	GMT log was generated (for example: yyyy/mm/dd hh:mm:ss).
< user >	User associated with log [currently unused].
< user flag >	User-defined enumeration.
< var data len >	Name of bytes of variable data associated with log.
< virtual dev >	Virtual device on Sensor.

Setting Alerts

To set an alert for the rule, select the checkbox next to Alert. If alert is selected and the rule is matched, IDP places an alert flag in the Alert column of the Log Viewer for the matching log record.

Logging Packets

To log packets around the attack, select the checkbox next to Log Packets. You can capture up to 256 packets before the event and 256 packets after the event.

NOTE: To improve IDP performance, log only the packets after the attack.

Setting SNMP and Syslog

To send an SNMP trap to the SNMP Manager, select the SNMP Trap checkbox in the Log Actions tab. To send a syslog entry to a syslog server, select the syslog checkbox in the Log Actions tab.

Sending Email

To send an email to designated recipients, check the Send Email checkbox and specify recipients. To specify new email recipients, click the + button and enter the email address. The new email addresses appear in the window.

Setting a Script

Scripts are custom scripts that take information about a security event and process it in a specific way. You might want to have several scripts available to respond to different security events. Customize each script based on the event that triggers it. For example, you might want to know what attack object triggered the event, the source and destination addresses, and so on.

To use new scripts, place them in the scripts directory on the NSM Device Server: /usr/netscreen/DevSvr/var/scripts/. The new scripts now appear in the run script pull-down menu in the Configure Notification dialog box.

To specify a script to be run, check the Run Script checkbox and select the script from the Script to Run pull-down menu.

Passing Values to a Script

When a security event occurs, the values passed to a script are the same values used to generate the log record for the event. In the Log Viewer, these values appear in the log-record fields; in your custom script, you can determine how the values are processed and displayed. Standard in (stdin) is used to pass values to the run script.

Setting Sensors

You can use a single security policy to control multiple Sensors. For each rule, you can select the Sensors that will use that rule to detect and prevent attacks. When you install the security policy that the rule belongs to, the rule becomes active only on the Sensors you selected in the Install On column of the rulebase.

Reducing False Positives

A false positive, also known as a false alert, is any situation in which benign traffic causes an intrusion detection system (IDS) to generate an alert. Common culprits of false positives are overly sensitive, inaccurate detection methods and applications that do not follow protocol RFCs. A few false positives from your IDS are normal, especially when you are testing new security policies, but too many false positives can degrade performance and produce oversized log files.

IDP reduces false positives by using *stateful* signatures to detect known attacks. A stateful signature knows the pattern and location of the attack, and produces fewer false positives than regular attack signatures because it eliminates network traffic that cannot contain the attack. To further increase detection accuracy and reduce false positives, IDP uses:

- Flow tracking to correlate multiple TCP/UDP connections into a single flow to determine the validity of the traffic
- IP defragmentation and TCP reassembly to reconstruct fragmented traffic
- Protocol normalization to normalize traffic to a common format for analysis

An efficient security policy also reduces false positives; keep the following tips in mind:

- The more specific the rule, the fewer the false positives. If you use any source and destination in your rule, be sure you really mean it. Exempting network objects and/or services improves performance and reduces false positives.

- When selecting attack objects by group, be sure you really want to include every attack in that group. For example, if you are running an Apache webserver, you probably do not need to check for Microsoft IIS attacks.
- If you do not specify any attacks in the Attack column, the rulebase acts like a firewall and performs the specified action for all traffic for that connection.

Working with Rulebases

Each rulebase in the IDP system uses a specific detection method to identify and prevent attacks. Together, the rulebases provide a Multi-Method Detection (MMD) system that can thwart almost any attack. This section describes the rulebases and provides examples of how to create rules that use each rulebase's detection methods.

NOTE: For more information about basic rule design, see "Designing Rules" on page 58.

You can add, modify, disable, and delete rules in five rulebases: Traffic Anomalies, SYN-Protector, Network Honeypot, IDP, and Backdoor Detection. You can use the Exempt rulebase to prevent rules in the IDP rulebase from matching on specific source and destination pairs.

The IDP Sensor applies rules to a connection in the following rulebase order:

- Traffic Anomalies. Protects your network from attacks by using traffic flow analysis to identify attack patterns over multiple connections.
- SYN-Protector. Protects your network from SYN-floods by ensuring that the three-way handshake is performed successfully for specified TCP traffic. You can choose from three methods of SYN-flood protection: none, relay (SYN Cookie), and passive.
- Network Honeypot. Protects your network by impersonating open ports, alerting you to attackers performing port scans and other information-gathering activities. Attackers that attempt to communicate with an impersonated port trigger a Network Honeypot rule and predefined action.
- IDP. Protects your network from attacks by using signatures and protocol-anomaly attack objects to identify malicious activity and take action against it. attack objects are grouped by severity, protocol, and attack type.
- Exempt. Prevents false positives by excluding specific source and destination pairs from matching specific attacks in the IDP rulebase.
- Backdoor Detection. Protects your network from dangerous backdoors (such as Trojans) by using heuristics to detect interactive traffic. The rulebase looks at network-traffic patterns and dynamically learns from the packet transmissions which traffic is interactive.

Using the Traffic Anomalies Rulebase

Traffic-anomaly rules protect your network from attacks by using traffic-flow analysis to identify attacks that occur over multiple connections and sessions (such as scans)

.

NOTE: The Traffic Anomalies rulebase is a terminate match rulebase. All rules are considered to be terminate match rules. When IDP finds a match on a rule, it does not execute succeeding rules.

Before attempting to enter an unknown network, attackers often gather information about the network and analyze any weaknesses to help them choose the best attack method. A port or network scan is often the first reconnaissance performed. Attackers typically use a scanning tool that attempts to connect to every port on a single machine (port scanning) or connect to multiple IP addresses on a network (network scanning). By determining which services are allowed and responding on your network, attackers can gain valuable information about your network configuration.

To detect scans and other distributed network attacks, the Traffic Anomalies rulebase looks for patterns that indicate abnormal network activity. Attackers often use scanning tools to automate their port scans, allowing them to scan multiple ports quickly and efficiently. IDP can detect these scans by counting the number of ports scanned in a specified period. You can also set a session limit threshold, which defines the maximum number of sessions for a single host.

Detecting TCP and UDP Port Scans

To detect TCP and UDP port scans, set a port count (number of ports scanned) and the time threshold (the period that ports are counted) in seconds.

Example: Traffic Anomalies Rule

You want to create a Traffic Anomalies rule that looks for port scans on your internal network. You set both the TCP and UDP Port Count to 20 and the Time threshold to 120 seconds. The rule is matched if the same source scans 20 TCP ports on your internal network within 120 seconds or if the same source scans 20 UDP ports on your internal network within 120 seconds.

Detecting Other Scans

In addition to port scans, the attacks can occur over multiple connections and sessions:

- Distributed Port Scans. Use multiple source addresses to scan ports.
- ICMP Sweeps. Use a single source to ping multiple IP addresses.
- Network Scans. Use a single source to scan multiple IP addresses.

To detect these attacks, set the IP Count (the number of times attempts to scan or ping ports on your network occur) and the Time (the period that IP addresses are counted) in seconds.

NOTE: Because a distributed port scan uses multiple source addresses, the source address appears as 0.0.0.0 in a scan log.

Example: Traffic Anomalies Rule

To create a Traffic Anomalies rule that looks for distributed port scans on your internal network, set the IP Count to 50 and the Time to 120 seconds. If 50 IP addresses attempt to scan ports on your internal network within 120 seconds, the rule is matched.

Example: Traffic Anomalies Rule

You want to create a Traffic Anomalies rule that looks for network scans and ICMP sweeps on your internal network. You set the IP Count to 50 and the Time to 120 seconds for ICMP sweeps and network scans. The rule is matched if:

- The same source attempts to scan 50 IP addresses on your internal network within 120 seconds.
- The same source attempts to ping 50 IP addresses on your internal network within 120 seconds.

Session Limiting

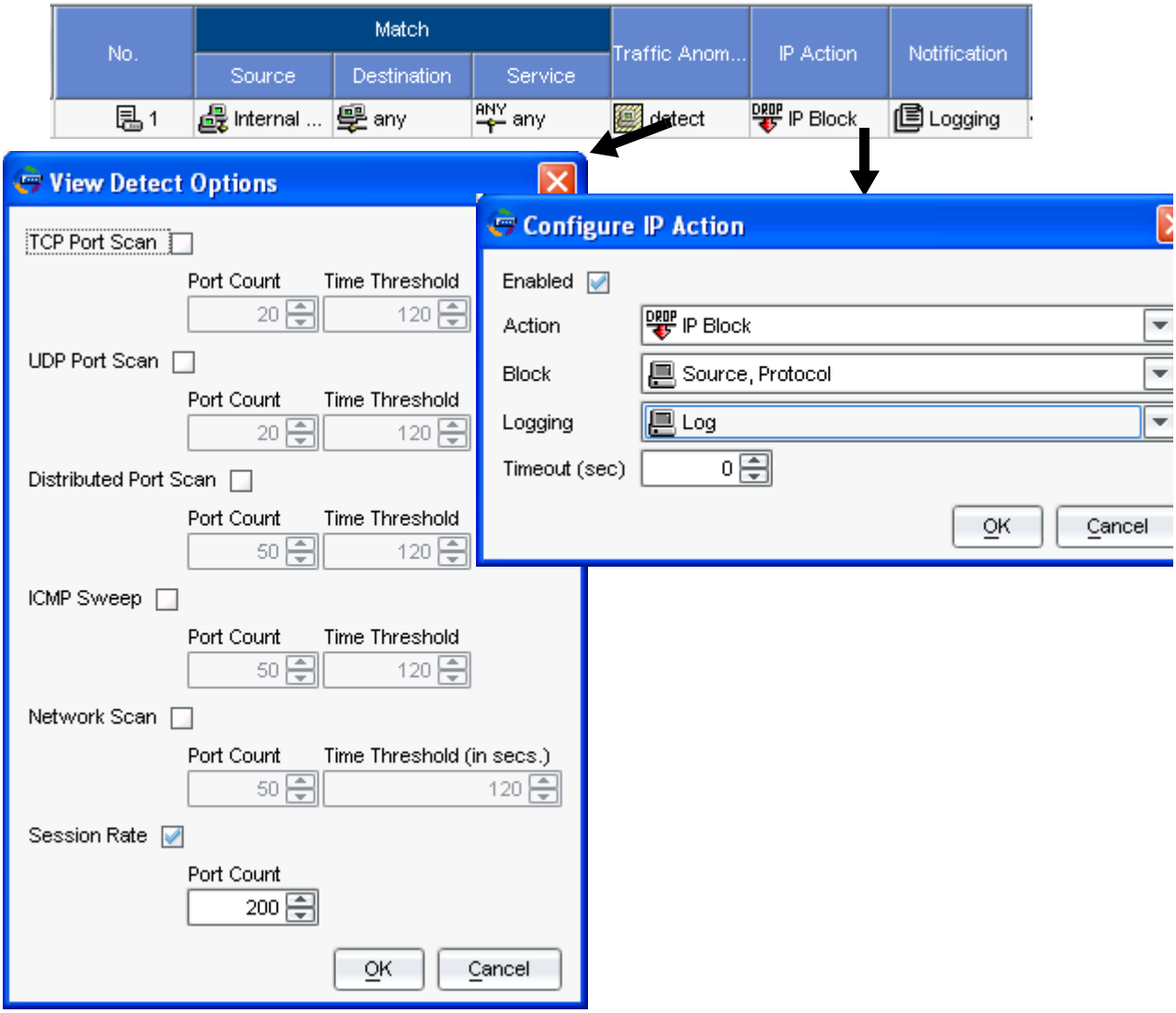
You can set a session-limit threshold that defines the maximum number of sessions allowed from a single host within one second. For each source specified in the rule, the Sensor tracks the sessions per second; if the session rate exceeds the user-defined maximum, the Sensor generates a SCAN_SESSION_RATE_EXCEEDED event log record, which appears in the Log Viewer. To take action when this event is triggered, configure an IP action in the rule.

NOTE: When you configure a Traffic Anomalies rule to block traffic that exceeds a specified session limit, the sessions counted are based on the source address only; the destination address and destination port values are set to 0. This means that if you set the blocking options for this rule to include the destination address or the destination port, IDP does not block traffic, as there are no sessions that match the destination address 0 or the destination port 0. When configuring session limiting for a Traffic Anomalies rule, you should use blocking options that specify the source address only and not the destination address/port. For example, select Source or Source, Protocol from the blocking options.

Example: Session Limiting

Your internal network typically has a low volume of traffic. To detect a sudden increase in traffic from a specific host (which might indicate a worm), set the source to your internal network and configure the session count as 200 sessions/second. To block traffic that exceeds the session limit, set an IP action of IDP Block and choose Source, Protocol from the Blocking Options menu. Your rule looks similar to this example:

Figure 12: Session Limiting Example



Creating a Traffic Anomalies Rule

Use the following guidelines when creating a Traffic Anomalies rule:

- Source and Destination. Set the source to any and the destination to the network objects you want to protect.
- Detect. Configure the Port Count and Time for TCP and UDP Port Scans. Configure the IP Count and Time for Distributed Port Scans, ICMP Sweeps, and Network Scans. Set the session limit, if desired.
- IP Action. Select the IP action that is appropriate for your network.

Using the SYN-Protector Rulebase

The SYN-Protector rulebase protects your network from SYN-floods by ensuring that the three-way handshake is performed successfully for specified TCP traffic. If you know that your network is vulnerable to a SYN-flood, use the SYN-Protector rulebase to prevent it.

NOTE: The SYN-Protector rulebase is a terminate match rulebase. All rules are considered to be terminate match rules. When IDP finds a match on a rule, it does not execute succeeding rules.

The TCP Handshake

When a TCP connection is initiated, a three-way handshake takes place:

- A client host sends a SYN packet to a specific port on the server to request a connection.
- Next, the server sends the client host a SYN/ACK packet, which both acknowledges (ACK) the original SYN packet from the client host and forwards a new SYN packet. The potential connection is now in a SYN_RECV state.
- Finally, the client host sends an ACK packet to the server to acknowledge receipt of the SYN/ACK packet. The connection is now in an ESTABLISHED state.

This three-way handshake contains an inherent, exploitable vulnerability that attackers can use to disable the system: a SYN-flood. Most systems allocate a large but finite number of resources to a connection table that is used to manage potential connections. While the connection table can sustain hundreds of concurrent connections across multiple ports, attackers can generate enough connection requests to exhaust all allocated resources.

SYN-Floods

Attackers initiate a SYN-flood by manipulating the basic three-way handshake:

- A client host sends a SYN packet to a specific port on the server. However, the attacker ensures that the client host's IP address is a spoofed IP address of an unreachable system.

- Next, the server sends the client host (spoofed address) a SYN/ACK packet. The potential connection is now in a SYN_RECV state.

- Since the system is unreachable, the server never receives an ACK or RST packet back from the client host. The potential connection is now in the SYN_RECV state and is placed into a connection queue while it waits for an ACK or RST packet. This potential connection remains in the queue until the connection-establishment timer expires (when it will be deleted).

- The attacker sends another SYN packet to the server, requesting another connection. And then another. And another. The connection table fills to capacity and cannot accept new SYN requests. The server is overwhelmed and quickly becomes disabled.

The SYN-Protector feature has two modes: passive and relay. As of IDP 4.0, relay mode makes use of SYN Cookies instead of keeping pending sessions in Sensor memory.

If the rule specifies Passive mode, then SYN protection is only activated when the number of SYN packets per second is greater than 1020. This number is the sum of two parameters that you can set in the Sensor Settings Run-Time Parameters:

- Lower SYN's-per-second threshold below which SYN-Protector will be deactivated (the default value is 1000).

- Upper SYN's-per-second threshold above which SYN-Protector will be activated (the default value is 20).

Once the SYN-Protector rulebase is activated, it remains active until the number of SYN packets per second is less than the Lower SYN's-per-second threshold (which is 1000 by default).

If the rule specifies relay (SYN Cookie) mode, then only the Lower SYN's-per-second threshold is used. If the threshold is exceeded, then SYN Cookies are added to subsequent SYN-ACK packets. If the level drops below, the Sensor drops out of SYN-Protector mode and lets the SYN packets reach the destination.

SYN-Flood Protection

SYN-flood protection can work in one of two modes:

- Passive: As SYN packets come in, the Sensor allocates resources to monitor the potential connections and passes the SYN packets to the destination. If SYN-flood thresholds have not been reached, the Sensor drops the uncompleted connections based on the normal TCP timeout. Once the thresholds have been reached, however, the Sensor uses a lower timeout value, which can be modified using the CLI. If the Sensor does receive a response, it passes the complete connection along to the destination and begins monitoring the session for anomalies.

- Relay: IDP 4.0 makes use of SYN Cookies. As SYN packets come in, but before the threshold has been reached, the Sensor allocates resources to monitoring the potential connections and passes the SYN packets along to the destination. Once the threshold is reached, the Sensor stops passing the SYN packets along, starts producing its own SYN-ACK packets with a SYN Cookie set, and stops

allocating resources to the potential connection. Once a valid ACK packet with SYN Cookie comes back in, the Sensor passes the connection along to the destination and allocates resources to monitoring the connection.

Creating a SYN-Protector Rule

Use the following guidelines when creating a SYN-protector rule:

- Source and Destination. To detect incoming interactive traffic, set the source to any and the destination to the network objects you want to protect from SYN-floods.
- Service. The default service, TCP-any, looks for SYN-floods in all TCP-based traffic.

NOTE: Always set the SYN-Protector service value to TCP-any. Selecting individual services can cause unpredictable interactions with other rulebases.

- Mode. Select the mode that indicates how IDP will handle TCP traffic:
  - None. IDP takes no action and does not participate in the three-way handshake.
  - Relay. IDP acts as the middleman, or relay, for the connection establishment, performing the three-way handshake with the client host on behalf of the server. Relay mode guarantees that the server allocates resources only to connections that are already in an ESTABLISHED state. The relay is transparent to both the client host and the server.

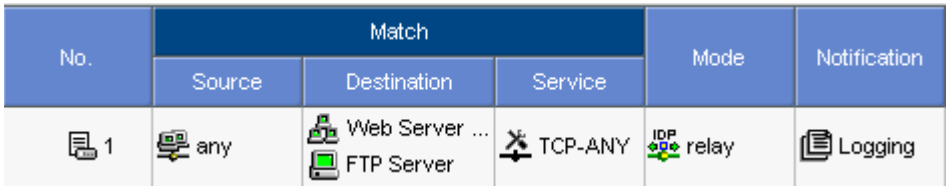
    IDP receives the initial SYN packet sent by the client host and returns a SYN/ACK packet with a SYN Cookie. If the client host sends an ACK packet with the appropriate cookie, IDP completes the three-way handshake and allows the connection to move to an ESTABLISHED state. If IDP does not receive an appropriate ACK packet from the client host, as would be the case during a SYN-flood attack, IDP does not complete the three-way handshake, and the connection is not established.
  - Passive. IDP handles the transfer of packets between the client host and the server but does not actively prevent the connection from being established. Instead, IDP uses a timer to ensure that connections are established promptly, minimizing the use of server resources. The timer IDP uses for the connection establishment is shorter than the timer the server uses for the connection queue.

    IDP transfers the SYN packet sent by the client host to the server, then transfers the SYN/ACK packet sent by the server to the client host. If the client host sends an ACK packet to the server before the IDP connection timer expires, the connection is established. If the client host does not send an ACK packet to the server, as would be the case during a SYN-flood attack, the IDP connection timer expires. IDP resets the connection to free resources on the server.

If you are protecting a large number of network objects, you can create a rule that excludes the traffic and objects you do not want to protect, then create another rule that includes all traffic and objects for protection. For more information, see “Setting Terminate Match Rules” on page 62.

A sample SYN-Protector rulebase rule is shown in Figure 13.

Figure 13: Sample SYN-Protector Rulebase Rule



Using the Network Honeypot Rulebase

A network honeypot protects your network by impersonating open ports on existing servers on your network, alerting you to attackers performing port scans and other information gathering activities.

NOTE: The Network Honeypot rulebase is a terminate match rulebase. All rules are considered to be terminate match rules. When IDP finds a match on a rule, it does not execute succeeding rules.

Impersonating a Port

Attackers view ports as entry points into your network. You can create counterfeit ports on existing servers to trick attackers who are attempting to break into your network. A counterfeit port can appear to offer notoriously vulnerable services to make the port attractive to attackers.

You create a counterfeit port in the Network Honeypot rulebase by specifying an existing network object and choosing a port and service to impersonate. You can also set an IP action to perform against the source. If an attacker attempts to communicate with your counterfeit port, the rule matches and the IP action is triggered. For more information about IP actions, see “Using IP Actions Against Existing Connections” on page 70.

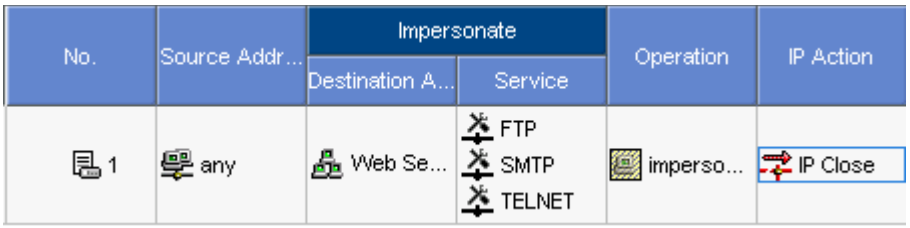
Creating a Network Honeypot Rule

Use the following guidelines when creating a Network Honeypot rule:

- Attacker IP. Set the Attacker IP to any.
- Destination. Select a network object, then choose a service and port that will appear to be available on this server.
- Operation. Set the Operation to impersonate.
- IP Action. Choose an IP Action that is appropriate for your network.

A sample Network Honeypot rule is shown in Figure 14.

Figure 14: Sample Network Honeypot Rule



Using the IDP Rulebase

The IDP rulebase protects your network from attack by using signature attack objects and protocol-anomaly attack objects to identify malicious activity and take action.

Creating an IDP Rulebase Rule

Use the following guidelines when creating an IDP rulebase rule:

- In the Match columns, specify the type of network traffic you want IDP to monitor for attacks:
  - *Source.* Select the source from the list of network objects, or set to any to monitor network traffic originating from any IP address. You can also negate a network object to specify all source addresses except those represented by the network object.
  - *Destination*. Select the destination from the list of network objects, or set to any to monitor network traffic sent to any destination. Selecting specific network objects for the destination enhances performance immensely by allowing IDP to monitor a subset of all network traffic. You can also negate a network object to specify all destination addresses except those represented by the network object.
  - *Service.* Specify the services that are supported by the destination you want IDP to monitor. The Service column is only visible if you select View > Show Expanded Mode.
  - *Terminate Match.* By default, rules in the IDP rulebase are non-terminal. You can set your rule to be terminate match. When a match is discovered against an attack object in a terminate match rule, the Sensor does not apply subsequent rules to that connection. For more information on terminate match rules and examples, see “Setting Terminate Match Rules” on page 62.
- In the Look For Attacks column, specify the attacks you want IDP to match in the monitored network traffic. Each attack object represents a known pattern of attack; when this known pattern of attack is encountered in the monitored network traffic, the attack object is matched. You can add attack objects by severity, by protocol, or by attack type; or you can add objects individually. For more information, see “Setting Attack Objects” on page 64.

- In the Action, IP Action, Notification, and Severity columns, specify the actions you want IDP to take when monitored traffic matches the rule's attack objects:
  - *Action*. Choose the action you want IDP to perform against the connection.
  - *IP Actions*. Choose the IP Actions that are appropriate for your network.
  - *Notification*. Choose none, or enable logging and select the logging options that are appropriate for your network.
  - *Severity*. Use the default severity setting of the selected attack objects, or choose a specific severity for your rule.

A sample IDP rulebase rule is shown in Figure 15.

Figure 15: Sample IDP Rulebase Rule

Using the Exempt Rulebase

The Exempt rulebase contains rules that prevent rules in the IDP rulebase from matching on specific source/destination pairs for specific attack objects. You might want to use an exempt rule under the following conditions:

- If your IDP rulebase rules use static or dynamic attack object groups containing one or more attack objects that produce false positives or irrelevant log records.
- To exclude a specific source, destination, or source/destination pair from attack detection.

The Exempt rulebase works in conjunction with the IDP rulebase: Before you can create exempt rules, you must first create rules in the IDP rulebase. If traffic matches a rule in the IDP rulebase, the IDP Sensor attempts to match the traffic against the Exempt rulebase before performing the specified action or creating a log record for the event.

You can create an exempt rule in the Exempt rulebase, or you can use the right-click menu in the Log Viewer.

NOTE: The Exempt rulebase is a non-terminal rulebase. IDP checks all rules in the Exempt rulebase and executes all matches.

Creating an Exempt Rule in the Exempt Rulebase

In your security policy, select the Exempt rulebase. Use the following guidelines when creating an Exempt rulebase rule:

- In the Match columns, specify the source and destination addresses for traffic you want to exempt:
  - *Source*. Select the source from the list of network objects, or set to any to exempt network traffic originating from any IP address. You can also negate a network object to specify all source addresses except those represented by the network object.
  - *Destination*. Select the destination from the list of network objects, or set to any to exempt network traffic sent to any destination. You can also negate a network object to specify all destination addresses except those represented by the network object.
- In the Attacks column, specify the attacks to exempt for the source/destination pair. If the IDP Sensor detects traffic that matches the specific source/destination pair and the attack objects in an IDP rulebase rule, the Sensor automatically exempts that traffic from attack detection.

You must include at least one attack object in your Exempt rule. For more information about configuring attack objects in rules, see “Setting Attack Objects” on page 64.

Example: Exempting a Source / Destination Pair

To improve performance and eliminate false positives between your Internal lab devices and your engineering desktops, you want to exempt attack detection. Your exempt rule looks similar to the example in Figure 16.

Figure 16: Sample Exempt Rule

No.	Match				Attacks
	From Zone	Source	To Zone	Destination	
1	any	Enginee...	any	Internal ...	All

Example: Exempting Specific Attack Objects

You consistently find that your security policy generates false positives for the attack HTTP Buffer Overflow: Header on your internal network. You want to exempt attack detection for this attack when the source is from your internal network. Your exempt rule looks similar to the example in Figure 17.

Figure 17: Exempting Specific Attack Objects

any	Internal ...	any	any	HTTP Buffer Overflow: Header

Using the Backdoor Detection Rulebase

The Backdoor Detection rulebase protects your network from dangerous backdoors (such as Trojans) by detecting interactive traffic. The rulebase looks at network traffic patterns and uses heuristics of packet transmissions to detect interactive traffic, a common sign of an attacker using a Trojan or backdoor.

NOTE: The Backdoor Detection rulebase is a terminate match rulebase. All rules are considered to be terminate match rules. When IDP finds a match on a rule, it does not execute succeeding rules.

Understanding Backdoors and Interactive Traffic

A backdoor is a mechanism installed on a host computer that facilitates unauthorized access to the system. Attackers who have already compromised a system can install a backdoor to make future attacks easier. When attackers send and retrieve information from a backdoor, they generate interactive traffic.

Interactive traffic indicates human involvement in a normally automated process, such as a user typing commands. Interactive traffic looks different from other traffic because humans are manually controlling one end of the connection. In a connection between two programs, the data transfer is automated; TCP packets can be batched and sent in bulk for efficiency. In a connection between a program and a user, packets are sent when they become available; characters display as they are typed (not after the word is complete). Interactive programs transmit several short IP packets containing individual keystrokes and their echoes, reflecting the real-time actions of a user (or an attacker).

Detecting Backdoors

When attackers type commands to control a backdoor, they generate interactive traffic that IDP can detect. Unlike antivirus software, which scans for known backdoor files or executables on the host system, IDP detects the interactive traffic that is produced when backdoors are used. This method ensures that IDP can detect all backdoors, both known and unknown. If interactive traffic is detected, IDP can perform IDP actions against the connection to prevent the attacker from further compromising your network.

Creating a Backdoor Detection Rule

Use the following guidelines to create a backdoor-detection rule:

- Source and Destination. To detect incoming interactive traffic, set the source to any and the destination to the network object you want to protect. To detect outgoing interactive traffic, set the source to the network object you want to protect and the destination to any.

- Service. Select interactive service objects. Be sure to include services that are offered by the source or destination as well as interactive services that are not; attackers can use a backdoor to install any interactive service. Do not include TELNET, SSH, RSH, NETMEETING, or VNC. These services are often used to remotely control a system legitimately and their inclusion might generate false positives.

- Action. Set the Operation to detect and choose an action to perform if interactive traffic is detected.

If you are protecting a large number of network objects from interactive traffic, you can create a rule that ignores accepted forms of interactive traffic from those objects, then create another rule that detects all interactive traffic from those objects. For more information, see “Setting Terminate Match Rules” on page 62.

Sample Backdoor Detection rulebase rules are shown in Figure 18.

Figure 18: Sample Backdoor Detection Rulebase Rules

No.	Match					Operation	Action
	From Zone	Source	To Zone	Destination	Service		
1	any	Web Server Group	any	any	ECHO FTP ntalk	Ignore	Accept
2	any	Web Server Group	any	any	any	Detect	Accept

Managing Security Policies

A security policy is the combination of all rulebases (and their rules) into a comprehensive plan that defines how the IDP system works on your network. Your goal is to design your security policy to be efficient and modular. Create multiple rules with a few attack objects each so you can disable individual rules or override the attack object severities.

If you are getting too many log records or log records that do not contain useful information, change the IDP actions to better suit your network traffic.

- Using the top-down method, you create a rule for each severity group and use the default actions. Next, install the security policy on the Sensors, analyze the results by viewing the log records and reports, then edit the security policy based on what you find. Identify false positives and eliminate the matching attack objects, then install the edited policy on the Sensors.

- Using the bottom-up method, you create one or two rules with a few individual attack objects that target your vulnerabilities. Next, install the security policy on the Sensors, analyze the results by viewing the log records and reports, and edit the security policy based on what you find. Identify false positives and eliminate the matching attack objects, add a few more individual attack objects to your existing rules, and create one or two more rules. Install the edited policy on the Sensors.

Remember that each network is unique. security policies that do not produce false positives on one network can produce them on another. You must experiment with your own network to determine how best to handle false positives. For information on tuning a security policy to your network, see “Fine-Tuning Security Policies” on page 23.

Using Security-Policy Templates

NSM includes several security-policy templates that contain rules for the IDP rulebase:

- all_with_logging. This template includes all attack objects and enable packet logging for all rules.
- all_without_logging. This template includes all attack objects but does not enable packet logging.
- dmz_services. Use this template to protect a typical DMZ environment.
- dns_server. Use this template to protect DNS services.
- file_server. Use this template to protect file sharing services, such as SMB, NFS, FTP, and others.
- getting_started. This template contains very open rules. It is a good place to start if you are just learning about IDP.
- idp_default. This template contains a good blend of security and performance.
- web_server. Use this template to protect HTTP servers from remote attacks.

Each security-policy template contains rules that use the default actions associated with the attack object severity and protocol groups. You should customize these security policies to work on your network by selecting your own network objects as the destination and choosing IDP actions that reflect your security needs.

Additionally, after you create a security policy, you can use that policy as a template for creating new security policies.

Verifying Security Policies

You can verify a security policy before you install it to identify potential problems. You should verify a security policy before installing it; a security policy that has internal problems can leave your network vulnerable to attacks.

The verification process identifies the following problems.

Rule Shadowing

Rule shadowing occurs when two rules are designed to detect the same attack. Eliminating rule shadowing improves IDP performance. To correct rule shadowing, return to the rule that is shadowing and modify or delete it.

Example: Rule Shadowing

You want to protect your internal network from HTTP attacks. Your rules look similar to this example:

No.	Match					
	From Zone	Source	To Zone	Destination	Service	Terminate Ma...
1	any	Europe Mail Server	any	any	HTTP	☑
2	any	Europe Mail Server	any	Internal Network	HTTP	☐

- Rule 1 matches all TCP-HTTP traffic from your European email server to any destination on your network.
- Rule 2 matches TCP-HTTP traffic from your European email server to your internal network.

Rule 2 shadows Rule 1 because any HTTP attacks on the internal network have already been detected by Rule 1.

Protocol Mismatches (IDP Rulebase Only)

Protocol mismatches occur when a service object that is specified in the Service column of the security policy uses a different protocol from that specified by the default service binding of the attack object for that rule.

Remember that the service binding specifies the service and port that the attack uses. Because two different protocols are specified, IDP cannot match attacks for the attack object. To correct, return to the rule and set the Service column to default.

Example: Protocol Mismatches

You want to protect your FTP server from UDP and FTP protocol attacks. Your rule looks similar to this example:

The default service binding (the protocol and port that the attack uses) for Critical FTP attack objects is FTP/21, which conflicts with the specified service object UDP/21

.

NOTE: You can create service objects for protocols that use nonstandard *ports*, but you cannot match attack objects to *protocols* they do not use.

Any-Any-None Rules (IDP Rulebase Only)

Any-Any-None rules are rules that specify any for the source and destination and none for Attacks. Because IDP must log all packets for all connections, this rule can cause severe IDP performance penalties. To correct, return to the rule and specify network objects for the destination and attack objects for the attacks.

Any-Any-One Rules (IDP Rulebase Only)

Any-Any-One rules are rules that specify any for the source and destination and a single attack object for attacks. Because IDP must look at all network traffic, this rule can cause severe IDP performance penalties. To correct, return to the rule and specify network objects for the destination.

Sniffer-Mode Restrictions

An IDP running in Sniffer mode cannot perform preventative actions on network traffic. Because IDP is not in-line, in the path of packets, it cannot take any action that affects the connection. A security policy for a Sniffer-mode IDP should not contain any of the following settings:

- Any rule with the IDP actions drop or drop_packet
- SYN-Protector rulebase rules that use Relay mode
- Anti-spoofing

No errors are generated if you decide to select these actions or detection mechanisms in your rule. However, because the IDP is not in-line, it cannot perform these actions or prevent attacks using these detection mechanisms. If a sniffer rule that has preventative actions is matched, the IDP system generates a log record with the action Dismiss in the Log Viewer Action column (if logging is enabled).

NOTE: A Sensor in Sniffer mode can send an RST to reset the connection when the action is Close Client and Server, Close Client, or Close Server. You must use the Appliance Configuration Manager (ACM) to configure a dedicated interface on the Sensor for this purpose.

To correct, return to the rule that created the problem and select the actions that monitor or log activity.

Installing Security Policies

After you have successfully verified your security policy, you can install it on your Sensors. For each rule, select the Sensors that will use that rule to detect and prevent attacks. When you install the security policy that the rule belongs to, the rule becomes active only on the Sensors you selected in the Install On column of the rulebase. A Sensor can only use one security policy at a time. When you install a new security policy, it overwrites any existing policies on the Sensor.

Disabling Rules

You can disable a rule in your security policy to prevent that rule from being applied to your network traffic. To disable a rule, right-click in the No. column of the rule and select Disable. Disabled rules still appear in the UI, but contain diagonal lines to help you identify them.

A rule is also disabled automatically when the rule does not contain at least one network object in the Source and Destination columns, at least one service object in the Service column, and at least one attack object in the Attacks column. This situation can occur when you delete objects from the Object Manager or when an attack object update removes all attack objects from a dynamic group used in the rule.

Example: Disabled Rule

Your security policy detects Microsoft Trojans, viruses, and worms being launched against your network. Your rule looks similar to this example:

You later decide that this rule is too broad, so you add more specific rules and disable the too-general rule. The rule looks similar to this example:

Printing and Exporting Rulebases

You can export (as PDF or PostScript) rulebases in your security policy. Refer to the NSM *Online Help* for details.

Chapter 8

Managing Attack Objects

This chapter covers the following topics:

- Signature Attack Objects
- Stateful Signatures
- Viewing and Editing Signature Attack Objects
- Working with Protocol-Anomaly Attack Objects
- Working with Compound Attack Objects
- Creating Attack Object Groups
- Updating the Attack Object Database
- Searching Attack Objects

Attack objects detect known and unknown attacks against your network. The IDP system includes a database of several hundred attack objects (signature attack objects and protocol-anomaly attack objects) that detect these attacks. Signature attack objects detect known attacks, and protocol-anomaly attack objects detect unknown attacks:

- Detecting Known Attacks (Signature Attack Object). Known attacks are the easiest to detect because you know what you are looking for. The world's security community, comprised of network administrators, security specialists, and security organizations (such as CERT), provide information on known attacks that can help you detect them. When a new attack is discovered, the security community analyzes the attack and reports its findings on how the attack works, where to look for it, and how to detect it.

  Even if you are not active in the security community, you have probably heard of attacks like the CodeRed worm or the ILoveYou virus in the media. Juniper Networks has created signature attack objects to detect these and other known attacks based on security community knowledge.

- Detecting Unknown Attacks (Protocol Anomaly Attack Objects). Unknown attacks are more complicated to detect because they are unknown—and unpredictable. You do not know what you are looking for or even where the attack might occur, so you need to be suspicious of any unusual activity on your network. You can monitor all traffic, all the time, hoping to catch something

amiss in your log records—or you can use protocol anomalies to detect suspicious traffic.

Signature Attack Objects

Signature attack objects detect known attacks using attack *signatures*. An attack signature is a pattern that always exists within an attack; if the attack is present, so is the attack signature. Think of the attack signature (also just called the signature) as a fingerprint. If an attack is in a message, it leaves a fingerprint (signature).

Unfortunately, most attacks do not simply write out their signature for you. To discover the attack signature, you must analyze the attack to detect a pattern within it. This pattern can be a specific segment of code, a URL, a value in a packet header, and so on. When you identify a pattern, you discover the signature of the attack, and therefore the means of identifying the attack itself.

Thousands of known attacks exist today, each with one or more attack signatures that identify them. However, signatures, just like your handwritten signature, often exist without context. A signature attempting to detect its own pattern within all network traffic might find a match in legitimate traffic.

This occurrence is called a false positive, which means that the signature has matched a pattern that is not an attack. False positives are bad because they distract you from the real attacks. To reduce false positives, a signature needs to look only at traffic that can contain the attack.

Stateful Signatures

IDP creates and uses *stateful* signatures to detect attacks. A stateful signature is a signature that not only knows the pattern it is attempting to find, but also knows where to look for that pattern. Stateful signatures produce very few false positives because they understand the context of the attack and can eliminate huge sections of network traffic they know the attack would not be in.

Stateful signatures are much smarter than regular signatures: they know the protocol or service used to perpetrate the attack, they know the direction and flow of the attack, and they know the context in which the attack occurs. Obviously, though, a signature cannot contain all this information within the attack signature pattern—the data must be associated with the signature, but not actually part of the pattern itself. IDP does this by combining the attack pattern with service, context, and other information into a signature attack object.

The IDP system includes several hundred signature attack objects that all use stateful signatures to detect known attacks. You can view, create, edit, or delete signature attack objects using NetScreen-Security Manager.

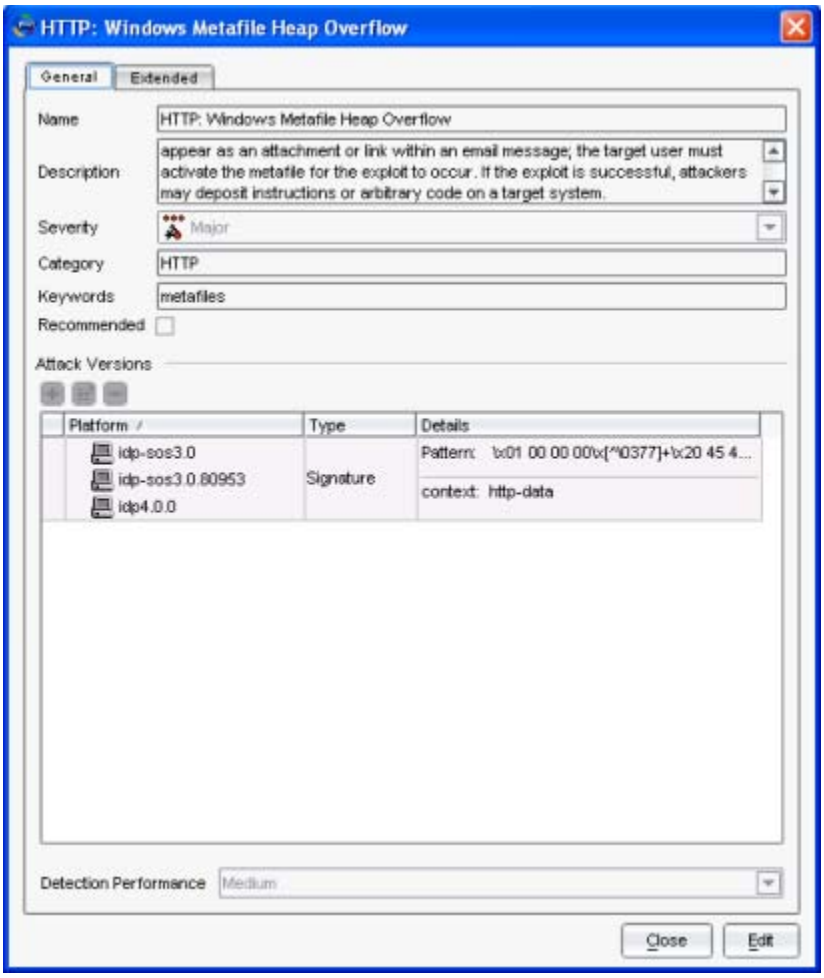
.

NOTE: For information about how to use signature attack objects in your rules, see “Setting Attack Objects” on page 64.

Viewing and Editing Signature Attack Objects

You can view, edit, and create signature attack objects. This section covers viewing and editing. The next section covers creating attack objects.

Figure 19: Attack Object Viewer



You might want to view an attack object to find out more about the attack it detects and the signature it uses. Or you might want to make a copy of an attack object and edit the context so that it produces fewer false positives. (You cannot edit predefined attack objects, but you can create copies of them to edit.)

All attack objects, including the ones you create yourself, are stored in the attack object database. Each signature attack object window divides its information into tabs. Some tabs contain subtabs.

- The General tab in the main dialog specifies the name, general description, severity, category, and keywords for the attack, plus whether or not the attack is recommended.

  This tab also contains a list of the Platforms the attack can be assigned to. Double-clicking the Platform entry brings up the Attack Version dialog, which contains two tabs. Different attack object types contain different information:

  Simple signature attack objects contain these tabs in the Attack Version dialog:

  - The General tab in the Platforms dialog specifies the platforms and type of the attack, how likely the attack object will generate false positives, the pattern and context of the attack, and service and time binding information about the attack.
  - The Header Match tab in the Platforms dialog contains two subtabs:
    - The IP tab in the Header Match tab covers IP-related settings.
    - The Protocols tab covers header matches for protocols.

  Compound attack objects contain these tabs in the Attack Version dialog:

  - The General tab in the Platforms dialog specifies the platforms and type of the attack, how likely the attack object will generate false positives, and service and time binding information about the attack.
  - The Compound Attack Members tab specifies the scope of the attack, the members of the compound attack, and the Boolean Expression and Ordered Match controls. These last two govern how the compound attack members are matched with the potential attack.

  Protocol-anomaly attack objects contain no tabs, just the Attack Version dialog:

  - The Protocol Anomaly dialog contains platform and false positive information, the pattern of the anomaly, and time binding information.

- The Extended tab in the main dialog displays background information about the attack, if available.

You access these tabs by double-clicking a particular attack object. Predefined attacks cannot be modified, only viewed. However, you can use the Edit button on a predefined attack to make a copy of the attack object. You can then modify and use the copy.

General Tab

The General tab specifies the attack properties and the attack pattern that IDP uses to match attacks.

Name

Name is used to display the attack object in most components of the UI.

When creating a new signature attack object, you might want to include the protocol the attack uses in the name for easy reference.

Description

Description provides details about the attack that the object detects, such as:

- Attack type (buffer overflows, password exploits, format string attacks, Denial-of-Service, and so on)
- Affected system (hardware, operating system, software application, or protocol the attack targets)
- Attack mechanism (how the attack works)
- Attack lethality (the consequences of a successful attack)

You are not required to include all this information when creating a new signature attack object, but it is a good idea. If you ever need to edit this signature, the description can help you remember important information about the attack.

Severity

Severity indicates how dangerous the attack is. IDP has five severity levels:

- Critical
- Major
- Minor
- Warning
- Info

Critical attacks are the most dangerous—typically these attacks attempt to crash your server or gain control of your network. Info (informational) attacks are the least dangerous and typically are used by network administrators to discover holes in their security system.

When you create a new signature attack object, choose a severity that matches the lethality of the attack. For more information about attack object severities, see “Adding Attack Objects by Severity” on page 65.

Category

Category can be used to lump similar types of attacks. Categories can include CHAT, HTTP, TELNET, FTP, TROJAN, WORM, and the like.

Keywords

Keywords indicate the important words that relate to the attack and the attack object. When creating a new signature attack object, entering keywords in this field can help you find it later.

Recommended Checkbox

If the Recommended checkbox is selected, it indicates that the person who created (or last updated) the attack object believes that the attack is a serious threat and ought to be included in policies. If the attack object is predefined, then a checkmark in this box indicates that Juniper Networks recommends this attack object be included in all relevant policies. You can create attack object groups that only contain recommended attack objects, or you can ignore this setting.

Platforms Wizard

The Platforms wizard lets you configure the meat of the attack object.

Platform functions:

- To view platform settings, double-click the platform entry.
- To edit platform settings, click the Edit button, then double-click the platform entry in the newly created copy of the attack object.
- To add a new platform entry, click the Edit button, then click the + button in the newly created copy of the attack object.

The Platform wizard has the following pages and fields. If you edit an existing attack object, the fields may be in a slightly different order.

Target Platform and Type

Target Platform

A platform is a particular operating system running on a particular piece of hardware (or hardware family.) For standalone IDP Sensors, IDP 4.0 is considered the platform, regardless of which model of standalone IDP Sensor the software is running on.

Select the platforms for this particular attack object.

Type

Indicates whether this is a signature or compound attack object. Select Signature to create a signature attack object.

General Properties Window

This window indicates the general properties of the attack. The fields vary depending on the type of attack selected in the previous window.

False Positives

The false positive setting indicates the frequency (unknown, rarely, occasionally, frequently) that the attack object produces a false positive when used in a security policy. By default, all signature attack objects are set to unknown; as you fine-tune your IDP system to your network traffic, you can change this setting to help you track false positives.

Binding Tab

The Binding tab displays the service binding for a signature attack object. A service binding is simply the service that an attack uses to enter your network. You can specify service bindings only for signature attack objects that use a packet, first packet, stream, stream 256, or line context (the service is already specified for attack objects that use a service context).

When creating a signature attack object, you cannot specify a service binding if you selected a service context.

However, if you choose a packet, first packet, stream, stream 256, or line context, you should set a service binding for the attack—this helps to significantly improve the accuracy of the signature attack object and can improve IDP performance.

Any

If you are unsure of the correct service, choose any. IDP attempts to match the signature in all services. Attacks can use multiple services to attack your network. Attack objects that detect these attacks use a service binding of any, allowing IDP to detect the attack regardless of which service the attack chooses for a connection.

IP

If you are not sure of the correct service but you know the IP protocol type, you can bind the attack pattern to the IP type. You can specify the name of the protocol type or the protocol type number.

You can use an IP protocol binding with IP Header Matches (detailed in “IP Tab” on page 112) or with an attack pattern. However, when using an IP protocol type with the first packet context, the attack pattern must be left empty.

The supported protocol types are shown in Table 10.

Table 10: Protocol Types Supported by IDP

Protocol Name	Protocol Type Number
IGMP	2
IPIP	4
EGP	8
PUP	12
IDP	22
TP	29
IPV6	41
ROUTING	43
FRAGMENT	44
RSVP	46
GRE	47
ESP	50
AH	51
ICMPV6	58
NONE	59
DSTOPTS	60
MTP	92
ENCAP	98
PIM	103
COMP	108
RAW	255

ICMP, TCP, and UDP

Attacks that do not use a specific service might use a specific protocol to attack your network. Services are Application Layer protocols that must use a Transport Layer or Network Layer protocol; attack objects that detect these types of attacks use a service binding that matches the attack's designated protocol.

Some TCP and UDP attacks use standard ports to enter your network and establish a connection; the matching attack objects detect these attacks by monitoring traffic on that service port or ICMP ID.

RPC

Remote Procedure Call (RPC) is used by distributed processing applications to handle interaction between processes remotely. When a client makes a Remote Procedure Call to an RPC server, the server replies with a remote program. Each remote program uses a different program number. Attack objects can use the RPC program to detect attacks against the RPC protocol.

Service

Many attacks use a specific service to attack your network. Attack objects that detect these types of attacks use a matching service binding.

Table 11: Service Bindings

Service	Description	Default Port
AIM	AOL Instant Messenger	
Chargen	Chargen	TCP/19, UDP/19
DHCP	Dynamic Host Configuration Protocol	
Discard	Discard	TCP/9, UDP/9
DNS	Domain Name System	TCP/53, UDP/53
Echo	Echo	TCP/7, UDP/7
Finger	Finger Information Protocol	TCP/79, UDP/79
FTP	File Transfer Protocol	TCP/21, UDP/21
Gnutella	Gnutella	
Gopher	Gopher	
HTTP	Hypertext Transfer Protocol	TCP/80, UDP/80
ICMP	Internet Control Message Protocol	
IDENT	IDENT	TCP/113
IMAP	Internet Message Access Protocol	TCP/143, UDP/143
IRC	Internet Relay Chat	
LDAP	Lightweight Directory Access Protocol	
LPR	Line Printer Protocol	
MSN	Microsoft Instant Messenger	
NBName NBDS	NetBios Name Service	UDP/137 (NBName) UDP/138 (NBDS)
NFS	Network File System	
NNTP	Network News Transfer Protocol	
NTP	Network Time Protocol	
POP3	Post Office Protocol, Version 3	TCP/110, UDP/110
Portmapper	Portmapper	TCP/111
RADIUS	Remote Authentication Dial-In User Service	
REXEC	Remote Execution	
RLOGIN	Remote Login	TCP/513
RSH	Remote Shell Utility	
RTSP	Real Time Streaming Protocol	
SMB	Server Message Block	
SMTP	Simple Mail Transfer Protocol	TCP/25, UDP/25
SNMP	Simple Network Management Protocol	TCP/161, UDP/161
SNMPTRAP	SNMP trap	TCP/162, UDP/162

Service	Description	Default Port
SSH	Secure Shell	TCP/22, UDP/22
SSL	Secure Sockets Layer	
syslog	System Log	UDP/514
Telnet	Telnet TCP protocol	TCP/23, UDP/23
TFTP	Trivial File Transfer Protocol	
VNC	Virtual Network Computing	
Whois	whois	
YMSG	Yahoo! Messenger	

Time Binding Fields

The Time Binding fields allow you to identify an attack that repeats over time across sessions. Time attributes are bound to the attack object for one minute. The following time attributes are used together to identify an attack that repeats for a certain number of times:

- Scope specifies whether the counting of the attack is from the same source address, the same destination address, or a pair of IP addresses. If you select Source, IDP detects attacks from a given source address for the specified number of times, regardless of the destination address. If you select Destination, IDP detects attacks to a given destination address for the specified number of times, regardless of the source address. If you select Peer, IDP detects attacks between source and destination addresses of the sessions for the specified number of times.

- Count/Min specifies the number of times per minute that IDP detects the attack within the specified scope before an event is triggered.

Note that a given attack that is bound to multiple ports and triggering on different ports is still counted the same way. For example, an attack triggering on TCP port 80 and then triggering on TCP port 8080 is counted as two separate occurrences. Also note that it is possible for the same signature or protocol anomaly to be in different attack objects: one with time attributes and the other without. For example, the TCP protocol normally Segment Out of Window is harmless and is normally seen occasionally on the network; however, thousands of these anomalies between given peers is suspicious.

Click Next to see the Attack Pattern window.

Attack Pattern Window

Pattern

The attack pattern is the signature of the attack you want to detect. Remember that a signature is a pattern that always exists within an attack; if the attack is present, so is the signature. Existing signature attack objects (the ones that are already in the attack object database) use stateful signatures created by the security professionals at Juniper Networks.

When creating a new signature attack object, you must analyze the attack to detect a pattern (segment of code, a URL, a value in a packet header, and so on) and then use that pattern to create a signature.

IDP uses a syntax based on regular expressions to match signature patterns, as shown in Table 12.

Table 12: Patterns for Signatures

Pattern	Syntax
Direct binary match (octal)	\0 < octal-number >
Direct binary match (hexadecimal)	\X < hexadecimal-number > \X
Case-insensitive matches	\[< character-set\]
Match any symbol	.
Match 0 or more symbols	*
Match 1 or more symbols	+
Match 0 or 1 symbols	?
Grouping of expressions	()
Alternation - typically used with ()	|
Character range	[< start > - < end >]
Negation of range	[^ < start > - < end >]

NOTE: Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel and copyright by the University of Cambridge, England. The source software is available using FTP from the following website: ftp://ftp.csx.cam.ac.ui/pub/software/programming/pcre/.

Example: Syntax Matches

Some example syntax matches are shown in Table 13.

Table 13: Example Syntax Matches

Syntax	Matches...	Example
\X01 86 A5 00 00\X	The 5 specified bytes verbatim.	01 86 A5 00 00
(hello|world)	hello or world.	hello world
(hello|world) +	hello or world one or more times.	helloworld worldhello hellohello
\[hello\]	hello in a case-insensitive manner.	hElLo HEllO heLLO
[c-e]a(d|t)	Expressions that begin with c, d, or e, the second letter a, and end in d or t.	cad cat dad dat ead eat
[^c-e]a(d|t)	Expressions that begin with a letter other than c, d, or e, have the second letter a, and end in d or t.	fad fat zad
a*b + c	Any number of a characters followed by one or more b characters followed by a c.	bc abc aaaabbbbc

Classic Attack Pattern: Finger Bomb

The finger bomb is a common Denial-of-Service (DoS) attack that causes a server to repeatedly forward requests to the same system. The @ symbol instructs a finger daemon to forward the preceding request to the server name that follows the symbol. If that server name also contains an @ symbol, the request is forwarded again to the server that follows—in the case of the finger bomb, consecutive @ symbols cause the request to be sent repeatedly. If the finger request forwards enough requests, the server eventually runs out of memory or available network sockets and crashes.

A classic signature for the finger bomb attack is shown here:

```
jdoe@@@@@@@@@@@@@@@@@@@@nextserver
```

The signature attack object that detects the finger bomb attack uses the following signature pattern:

```
.*@@.*
```

Negate

Select the Negate checkbox if you want to detect packets that do *not* have the pattern indicated.

Context

The context defines the location of the signature. Remember that stateful signatures know both the attack pattern and where to look for that attack pattern. The context of a signature attack object is very important because it specifies where IDP looks for attacks. Signature attack objects with packet, first packet, line, stream 256, or stream contexts detect attacks that use an unspecified context. Signature attack object with a service context detect attacks that use a specific context.

When creating a new signature attack object, you can tell IDP to detect attacks in multiple contexts, or you can limit the search to just one context. Choosing just one or two contexts makes IDP more effective and reduces false positives:

- Packet context tells IDP to match the pattern within a packet. When creating a signature attack object that uses a packet context, you should also specify the packet service binding in the binding tab and define the packet header options in the IP tab. Although not required, specifying these additional parameters helps to improve the accuracy of the signature attack object, and can improve IDP performance.

- First data packet context tells IDP to monitor only the first data packet of a stream for a pattern match. When the flow direction for the attack object is set to any, IDP checks the first data packet of both the STC and CTS flows. If you know that the attack signature will appear in the first data packet of a session, choosing first data packet instead of packet will reduce the amount of traffic the IDP needs to monitor, improving performance.

- First packet context tells IDP to monitor only the first packet of a stream for a pattern match. When the flow direction for the attack object is set to any, IDP checks the first packet of both the STC and CTS flows. If you know that the attack signature will appear in the first packet of a session, choosing first packet instead of packet will reduce the amount of traffic the IDP needs to monitor, improving performance.

- Stream context tells IDP to reassemble packets and extract the data to search for a pattern match. However, IDP does not recognize packet boundaries for stream contexts, so data for multiple packets is combined. When creating a signature attack object that uses a stream context, you should also specify the stream service binding in the binding tab. Again, although not required, this helps to improve the accuracy of the signature attack object and IDP performance.

- Stream 256 context tells IDP to reassemble packets to search for a pattern match within the first 256 bytes of a traffic stream. When the flow direction for the attack object is set to any, IDP checks the first 256 bytes of both the STC and CTS flows. If you know that the attack signature will appear in the first 256 bytes of a session, choosing stream 256 instead of stream will reduce the amount of traffic the IDP needs to monitor and cache, improving performance.

- Stream 1K context tells IDP to reassemble packets to search for a pattern match within the first 1K bytes of a traffic stream. When the flow direction for the attack object is set to any, IDP checks the first 1K bytes of both the STC and CTS flows. If you know that the attack signature will appear in the first 1K bytes of a session, choosing stream 1K instead of stream will reduce the amount of traffic the IDP needs to monitor and cache, improving performance.

- Stream 8K context tells IDP to reassemble packets to search for a pattern match within the first 8K bytes of a traffic stream. When the flow direction for the attack object is set to any, IDP checks the first 8K bytes of both the STC and CTS flows. If you know that the attack signature will appear in the first 8K bytes of a session, choosing stream 8K instead of stream will reduce the amount of traffic the IDP needs to monitor and cache, improving performance.

- Line context tells IDP to look for a pattern match within a specific line within your network traffic.

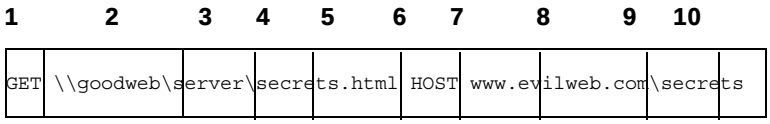
Example: Context HTTP URL Attack (Part 1 of 2)

You store important information about your webserver in the file directory C:\\goodweb\server\secrets.html. To protect this file, you create a signature attack object with the attack pattern \secrets. A malicious host, www.evilweb.com, sends your webserver the following command:

```
GET \\goodweb\server\secrets.html
```

Here is how each context option handles the command:

Packet context. The incoming command is fragmented into 10 packets for transfer over TCP. The first 5 fragments contain bytes for the command; the final 5 fragments contain bytes for the host. Because no single packet contains the string \secrets, IDP cannot match the attack to the attack pattern and does not detect the attack.



Stream Context. As before, the incoming command is fragmented and no single packet contains the attack pattern. Instead of looking for the attack in individual packets, IDP uses stream reassembly to extract HTTP data from all fragments and successfully recreates the original command. Stream reassembly does not recognize packet boundaries, so both the command and host data are combined into one stream of HTTP data:

```
GET \\goodweb\server\secrets.html HOST www.evilweb.com\secrets
```

IDP detects a match for the attack pattern in the command as shown here. However, IDP also detects a second match in the host name, creating one false positive.

Line Context. As in the packet context and stream context, IDP reassembles the fragmented packets and extracts the HTTP data stream. IDP separates the stream into individual lines, based on the line terminators included in the original HTTP header:

		Line Terminator
Line 1	`GET \\goodweb\server\secrets.html`	`\r\n`
		Line Terminator
Line2	`HOST www.evilweb.com\secrets`	`\r\n`

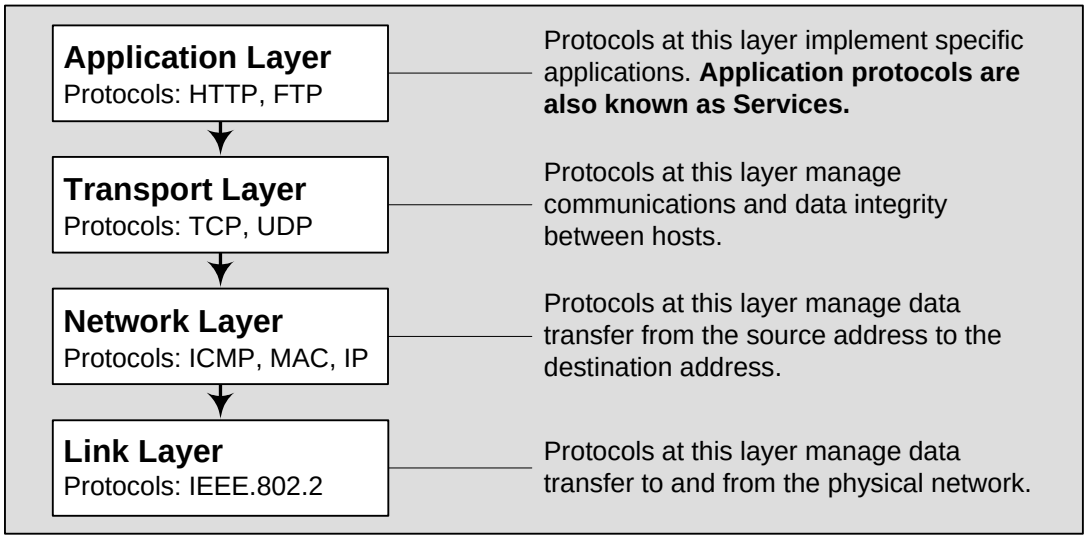
IDP successfully detects two matches for the attack pattern: one real attack in Line 1 and one false positive in Line 2.

Service Contexts

A service context tells IDP to look in a specific service context for the attack. To understand how service contexts work, it is important to know how networks use protocols to transfer data.

Protocols are a set of rules that determine how two (or more) computers communicate with each other. Because protocols often rely on other protocols to handle communication tasks that are outside their own functionality, a layered model works best for understanding protocol interaction. The TCP/IP Model of communication uses four protocol Layers: Link, Network, Transport, and Application. Each protocol Layer uses different protocols to manage a specific aspect of the communication process.

Figure 20: Protocol Layers



The highest protocol Layer, Application, contains service protocols. Services define how data is structured; a service context is a specific location within that defined data structure. A service can contain multiple service contexts. Services must still use a Transport or Network Layer protocol to actually transfer that data.

When creating a signature attack object, choose a service context if possible. Because the service context is very specific, your chances of detecting a false positive are greatly reduced. However, choosing a service context overrides the default service binding in the Binding tab. Service contexts can specify the exact location of an attack.

IDP supports contexts for the services shown in Table 14.

Table 14: Service Contexts

	Description	RFC
AIM	AOL Instant Messenger	
DHCP	Dynamic Host Configuration Protocol	2131, 2132
DNS	Domain Name System	1034, 1035
Finger	Finger Information Protocol	1128
FTP	File Transfer Protocol	959
Gnutella	Gnutella	
Gopher	Gopher	1436
HTTP	Hypertext Transfer Protocol	2616
IMAP	Internet Message Access Protocol	2060
IRC	Internet Relay Chat	2810, 2811, 2812, 2813
LDAP	Lightweight Directory Access Protocol	2251, 2252, 2253, 3377
LPR	Line Printer Protocol	1179
MSN	Microsoft Instant Messenger	
NBNAME NBDS	NetBios Name Service	1001, 1002
NFS	Network File System	
NNTP	Network News Transfer Protocol	977
NTP	Network Time Protocol	1305
POP3	Post Office Protocol, Version 3	1081
RADIUS	Remote Authentication Dial In User Service	2865, 2866, 2867, 2868, 3575
REXEC	Remote Execution	
RLOGIN	Remote Login (rlogin)	1258, 1282
RSH	Remote Shell (rsh)	
RUSERS		
SMB	Server Message Block	
SMTP	Simple Mail Transfer Protocol	821
SNMP	Simple Network Management Protocol	1067
SNMPTRAP	SNMP trap	1067
SSH	Secure Shell	Proprietary

	Description	RFC
SSL	Secure Sockets Layer	
Telnet	Telnet TCP protocol	854
TFTP	Trivial File Transfer Protocol	783
VNC	Virtual Network Computing	
YMSG	Yahoo! Messenger	

For details on Service contexts, refer to the NSM *Online Help* topic on creating and editing attack objects without packet context.

Example: Context HTTP URL attack (Part 2 of 2)

In part one of this example, we saw how IDP used packet, stream, and line context for the attack pattern \secrets to detect an attack in the following HTTP command:

```
GET \\goodweb\server\secrets.html.
```

While the line context successfully detected the attack, it also generated a large number of false positives (any requesting host with a secrets directory triggered a match). You can update the signature attack pattern to be more specific; you now want to detect only the attempts to access the first level secrets directory. Your attack pattern is now goodweb\server\secrets.html.

Now let us pretend that our malicious host, www.evilweb.com, has also discovered some new tricks and has returned to your network for another break-in attempt. Now the malicious HTTP command looks much more complex:

```
GET \\goodweb\server\anydir\..\.%c1%9csecrets.html
```

Like many savvy attackers, evilweb.com has discovered how to bypass IDS string-matching attempts by using directory traversals and unicode. Directory traversal uses relative paths to move up and down the webserver directory tree.

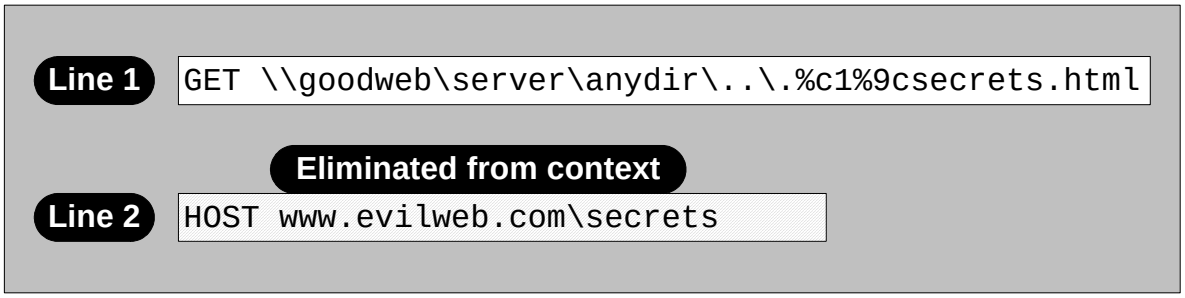
	down one directory
.\	same directory
..\	up one directory

A webserver automatically performs the directory traversal when executing the URL command. IDP must parse the URL command by dividing the command into components, identifying each component type, and analyzing the component before it can determine the appropriate action.

Unicode is a widely accepted standard that attempts to electronically map characters of every language (65,000) to a 16-bit number. “%c1%9c” is the unicode representation of “\”. A webserver, using a process called *normalizing*, automatically decodes unicode representations into their equivalent characters. IDP must parse the URL command before it can identify and normalize the unicode component.

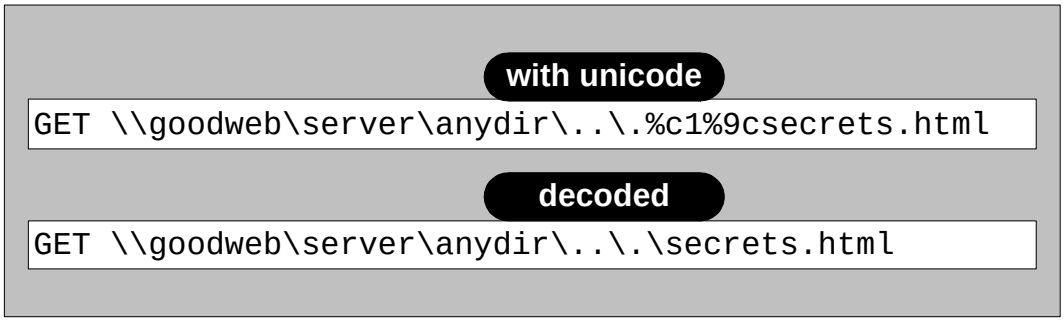
Here is how HTTP-URL and HTTP-URL-PARSED service contexts would handle this attack:

HTTP-URL. IDP recognizes the incoming command as an HTTP URL, allowing it to differentiate the command line from the host line without the aid of line terminators. Because IDP is looking for the attack only in the HTTP-URL context, it eliminates the host line as irrelevant, as shown here:

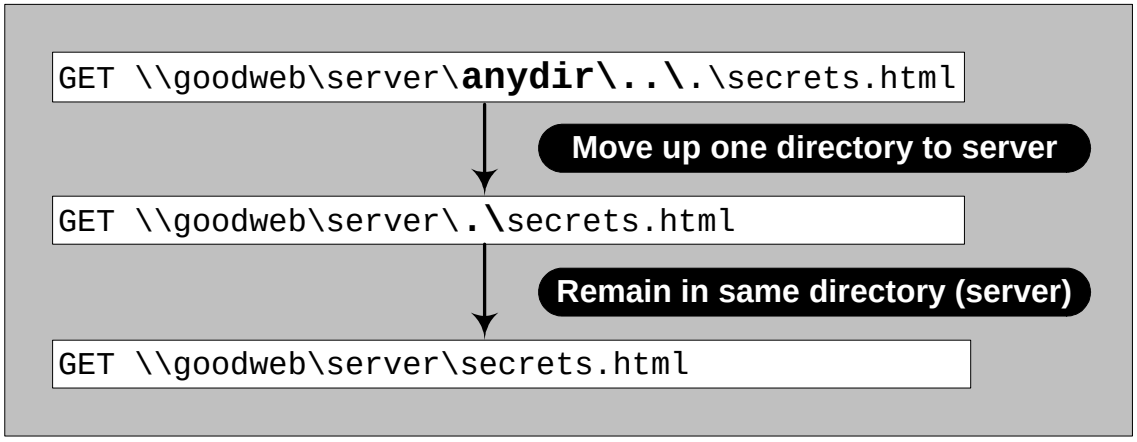


IDP cannot match the attack pattern, goodweb\server\secrets, to this new command because of the intervening directories and unicode. To detect this attack, IDP must parse the incoming command.

HTTP-URL-PARSED. As before, IDP recognizes the incoming command as an HTTP URL and eliminates the host line as irrelevant. IDP parses the command line and normalizes the unicode, as shown here:



Then IDP performs the directory traversal, as shown here:



IDP successfully matches the attack pattern to the parsed URL.

Direction

The direction defines the connection direction of the attack:

- Client to Server detects the attack only in client-to-server traffic.
- Server to Client detects the attack only in server-to-client traffic.
- Any detects the attack in either direction.

When creating a new signature attack object, choose a single direction to improve IDP Sensor performance, reduce false positives, and increase detection accuracy.

Flows

The flow defines the connection flow of the attack:

- Control detects the attack in the initial connection that is established persistently to issue commands, requests, and so on.
- Auxiliary detects the attack in the response connection established intermittently to transfer requested data.
- Both detects the attack in the initial and response connections.

When creating a new signature attack object, choose a single flow to improve IDP Sensor performance and detection accuracy.

Click Next to see the IP Settings and Header Match page.

IP Tab

The IP tab specifies the contents of the IP header in a malicious packet. You cannot specify IP header contents if you selected a line, stream, stream 256, or a service context in the Basic tab. However, if you selected a packet or first packet context, you can define IP header contents for the malicious packet. If you are unsure of the IP flags and IP fields for the malicious packet, leave all fields blank and IDP will attempt to match the signature for all IP header contents.

IP Fields

You can set values for the following IP fields:

- Type of Service is one of the following service types:
  - 0000 Default
  - 0001 Minimize Cost
  - 0002 Maximize Reliability
  - 0003 Maximize Throughput
  - 0004 Minimize Delay
  - 0005 Maximize Security
- Packet Length is the number of bytes in the packet, including all header fields and the data payload.
- ID is a unique value used by the destination system to reassemble a fragmented packet.
- Time to Live is the time-to-live (TTL) value of the packet. This value represents the number of routers the packet can pass through. Each router that processes the packet decrements the TTL by 1; when the TTL reaches 0, the packet is discarded.
- Protocol is the protocol of the packet (ICMP, TCP, and so on).
- Source is the IP address of the attacking device.
- Destination is the IP address of the attack target.

IP Flags

You can also specify that IDP look for a pattern match whether or not a certain IP flag is set (none), only if the flag is set (set), or only if the flag is not set (unset):

- RB (Reserved Bit). This bit is not used.
- MF (More Fragments). When set (1), this option indicates that the packet contains more fragments. When unset (0), it indicates that no more fragments remain.
- DF (Don't Fragment). When set (1), this option indicates that the packet cannot be fragmented for transmission.

Protocols Tab

The Protocols tab allows you to specify that IDP search a packet for a pattern match only if certain protocol header information is set for TCP, UDP, or ICMP. This tab appears only for signature attack objects that use a packet or first packet context.

You can specify different options for TCP, UDP, and ICMP protocol headers, or choose None to match all fields in all protocol headers. For each protocol field value you enter, you must also specify the relational or equality operator: equal to, greater than, less than, and so on.

TCP Header Matches

For TCP, you can set values for the following TCP fields:

- Source Port is the port number on the attacking device.
- Destination Port is the port number of the attack target.
- Seq.Number. The sequence number of the packet. This number identifies the location of the data in relation to the entire data sequence.
- ACK Number is the ACK number of the packet. This number identifies the next sequence number; the ACK flag must be set to activate this field.
- Header Length is the number of bytes in the TCP header.
- Data Length is the number of bytes in the data payload. For SYN, ACK, and FIN packets, this field should be empty.
- Window Size is the number of bytes in the TCP window size.
- UrgPtr (Urgent Pointer) indicates that the data in the packet is urgent; the URG flag must be set to activate this field.

You can also specify the following TCP flag options as none, set, or unset:

- Urgent bit. When set, the urgent flag indicates that the packet data is urgent.
- ACK bit. When set, the acknowledgment flag acknowledges receipt of a packet.
- PSH bit. When set, the push flag indicates that the receiver should push all data in the current sequence to the destination application (identified by the port number) without waiting for the remaining packets in the sequence.
- RST bit. When set, the reset flag resets the TCP connection, discarding all packets in an existing sequence.
- SYN bit. When set, indicates that the sending device is asking for a three-way handshake to initialize communications.
- FIN bit. When set, the final flag indicates that the packet transfer is complete and the connection can be closed.
- R1 bit. When set, indicates that the R1 retransmission threshold has been reached.

- R2 bit. When set, indicates that the R2 retransmission threshold has been reached.

UDP Headers

For UDP, you can set values for the following fields:

- Source Port. The port number on the attacking device.
- Dest. Port. The port number of the attack target.
- Data Length. The number of bytes in the data payload.

ICMP Headers

For ICMP, you can set values for the following fields:

- Type. The primary code that identifies the function of the request/reply.
- Code. The secondary code that identifies the function of the request/reply within a given type.
- Seq. Number. The sequence number of the packet. This number identifies the location of the request/reply in relation to the entire sequence.
- ID. The identification number is a unique value used by the destination system to associate requests and replies.
- Data Length. The number of bytes in the data payload.

Extended Tab

The Extended tab displays known network-security references for the attack.

Network-security references are links to some of the security community's official descriptions of an attack. Because the security community is continually discovering and analyzing new attacks, and communicating their findings to each other on the Internet, references to their work are often the best way to truly understand an attack.

The attack description in the basic tab provides a brief overview of the attack. However, if you want more detailed information about how an attack works, or want to research an existing attack to compare to a suspected new attack, check the references.

URLs

Enter URLs that provide more information about the attack in the Primary URL, Secondary URL, and Tertiary URL fields.

Standard References

- CVE (Common Vulnerabilities and Exposures). A standardized list of vulnerabilities and other information security exposures. Enter the CVE URL for the vulnerability.
- BugTraq. A moderated mailing list that discusses and announces computer security vulnerabilities. Enter the BugTraq number for the vulnerability.

More Info

- Impact. Details the impact if the attack is successful, including information on system crashes and access granted to the attacker.
- Description. Provides a history of the attack and detailed information about how the attack works and what products are affected.
- Tech Info. Provides details on the vulnerability, the commands used to execute the attack, which files are attacked, registry edits, and other low-level information.
- Patches. Lists available patches available from the product vendor, as well as information on how to prevent the attack.

NOTE: Additional information is not available for all signature attack objects.

Working with Protocol-Anomaly Attack Objects

Protocol-anomaly attack objects detect abnormal or ambiguous messages within a connection according to a set of default rules. The IDP system includes several hundred protocol-anomaly attack objects to detect unknown attacks. You can view protocol-anomaly attack objects, but you cannot create, edit, or delete them.

As discussed earlier, protocol anomalies are deviations from the standard protocol, and a protocol is simply a method of communicating that is well-known and understood. Most legitimate traffic adheres to the established protocols, but traffic that does not produces an anomaly.

Illegal or ambiguous traffic does not just happen—attackers create anomalies for a specific purpose, like evading an IDS.

NOTE: Some software applications produce legitimate traffic that can appear as protocol anomalies as a result of nonstandard implementation of a protocol RFC.

Detecting Protocol Anomalies

Protocol-anomaly attack objects are designed to detect unknown attacks, and can help you identify unusual activity on your network. IDP uses protocol-anomaly detection (also called *protocol analysis*) to determine illegal or ambiguous packets that can constitute security threats by checking them against the protocol RFCs or the definitions imposed by the network administrator.

IDP includes all known protocol anomalies as protocol-anomaly attack objects in the attack object database. IDP detects anomalies for the following protocols:

AIM	DHCP	IDENT	RUSERS	TFTP
FINGER	CHARGEN	IMAP	Gnutella	RLOGIN
FTP	DISCARD	IP Packet	Gopher	RPC
HTTP	DNS	POP3	IRC	RSH
ICMP	ECHO	REXEC	MSN	RTSP
MSN	LPR	NFS	VNC	NNTP
SNMP	SMTP	SMB	SNMP TRAP	YMSG
TCP segment	SYSLOG	SSH	TELNET	

You can use the Sensor Settings rulebase to adjust thresholds and configuration options for many of the included protocols. However, you cannot create, edit, or delete protocol-anomaly attack objects.

Viewing Protocol-Anomaly Attack Objects

The Protocol Anomaly Editor has two tabs of information that detail each protocol-anomaly attack object: Basic and Extended. Within the Basic tab is the Attack Version dialog.

Basic Tab

Name

Name is used to display the attack object in most components of the UI.

Description

Description provides details about the attack that the attack object detects, such as:

- What the anomaly is
- Affected system (hardware, operating system, software application, or protocol the attack targets)
- Attack mechanism (how the attack works)
- Attack lethality (the consequences of a successful attack)

You are not required to include all this information when creating a new signature attack object, but it is a good idea. If you ever need to edit this signature, the description can help you remember important information about the attack.

Severity

Severity indicates how dangerous the attack is. IDP has five severity levels:

- Critical
- Major
- Minor
- Warning
- Info

Critical attacks are the most dangerous—typically these attacks attempt to crash your server or gain control of your network. Informational attacks are the least dangerous, and typically are used by network administrators to discover holes in their own security system.

When you create a new signature attack object, choose a severity that matches the lethality of the attack. For more information about attack-object severities, see “Adding Attack Objects by Severity” on page 65.

Category

Category can be used to lump similar types of attacks. Categories can include CHAT, HTTP, TELNET, FTP, TROJAN, WORM, and the like.

Keywords

Keywords indicate the important words that relate to the attack and the attack object. When creating a new signature attack object, entering keywords in this field can help you find it later.

Recommended

If the Recommended checkbox is selected, it indicates that the person who created (or last updated) the attack object believes that this attack object covers a serious vulnerability. For predefined attack objects, Juniper Networks sets selected attack objects as Recommended.

Attack Version

The fields the Attack Version dialog are the same as the fields in the signature attack object, except for the Anomaly field. The Anomaly field indicates which known protocol anomaly the object detects.

Extended Tab

The Extended tab displays known network-security references for a protocol-anomaly attack object.

Working with Compound Attack Objects

A compound attack object enables you to be very specific about the events that need to take place before IDP identifies traffic as an attack. Use compound attack objects to refine your security policy rules, reduce false positives, and increase detection accuracy.

For example, you might want to take action only if an FTP session included a failed login attempt for specific users. You can view, edit, and create Compound attack objects, which combine multiple signatures and/or protocol anomalies into a single attack object. Traffic must match all of the combined signatures and/or anomalies to match the Compound attack object—you can even specify the order in which signatures or anomalies must match.

When creating a custom compound attack object, keep the following rules in mind:

- All members of the compound attack object must use the same service setting or service binding, such as FTP, Telnet, YMSG, TCP/80, and so on.
- You can add protocol-anomaly attack objects to a compound attack object.
- You *cannot* add predefined or custom signature attack objects to a compound attack object. Instead, you specify the signature directly within the compound attack object, including such details as service (or service binding), service context, attack pattern, and direction.
- You can add between 2 and 32 protocol-anomaly attack objects and/or signatures as members of the compound attack object. However, all members must use the same service setting or service binding.

Compound attack objects contain the same fields as a signature attack object, except that the Compound Members pane is different.

Compound Members Pane

The Compound Members pane lets you specify what the individual members will be and how they will interact.

Scope

If the selected service supports multiple transactions within a single session, you can also specify whether the match should occur over a single session or can be made across multiple transactions within a session:

- Select session to allow multiple matches for the object within the same session.
- Select transaction to match the object across multiple transactions that occur within the same session.

Ordered Match

Use ordered match to create a compound attack object that must match each member signature or protocol anomaly in the order you specify. If you do not specify an ordered match, the compound attack object still must match all members, but the attacks or protocol anomalies can appear in random order.

To configure an ordered match, select the Ordered Match checkbox and use the arrow keys to reorder members.

Configuring Boolean Expression

Using the Boolean Expression field override the Ordered Match function for standalone IDP devices.

The Boolean Expression field makes use of the Member Names created in the lower part of the dialog.

NSM supports three Boolean operators: or, and, and oand (ordered and). NSM also supports the use of parenthesis to determine precedence.

Boolean operators:

- or - if either of the member name patterns match, the expression matches.
- and - if both of the member name patterns mach, the expression matches. It does not matter which order the members appear in.
- oand - if both of the member name patterns match, and if they appear in the same order as in the Boolean Expression, the expression matches.

Example: Boolean Expression

Suppose you have created signature members labeled s1 - s5.

Suppose you know that the attack always contains the pattern s1, followed by either s2 or s3. Further, you know that the attack always contains s4 and s5, but their positions in the attack can vary.

You might create the following Boolean expression:

```
(s1 oand (s2 or s3)) and (s4 and s5)
```

Add Signature

To add an attack pattern to the compound attack object, click + , select Add Signature, and configure the Attack Pattern settings:

- Pattern. Specifies the pattern IDP should match. You construct the attack pattern just as you would when creating a new signature attack object. To negate the pattern, select the Negate checkbox.
- Context. Specifies the context in which the IDP should look for the pattern. The context displays only contexts that are appropriate for the specified Service. If you selected a service binding of *any*, you are restricted to the service contexts packet and first packet.
- Direction. Specifies whether IDP should match the pattern in traffic flowing in any direction, from client to server, or from server to client.

Add Anomaly

To add a protocol anomaly to the compound attack object, click + and select Add Anomaly. Give the member a name. In the Attack Properties area, select an anomaly from the Key menu. The menu only displays protocol anomalies appropriate for the service you selected. If you selected a service binding of any, you are restricted to the IP-based protocol-anomaly attack objects.

Delete

To remove a member signature or anomaly, select the member in the list, then click Delete. A confirmation window asks you to verify that you want to delete the item.

Creating Attack Object Groups

NSM contains hundreds of predefined attack objects, and you can create hundreds more using custom attack objects. To help keep your security policies organized, you should organize your attack objects into groups.

The IDP system organizes attack objects by severity (Critical, Major, Minor, Warning, and Info), by protocol, and finally by attack type. When you create your security policy rules, you can add these attack objects by individually, or by the predefined severity, protocol, or attack type group. While these groups are designed to protect your network, they also cover all attacks, many of which your network might not be vulnerable to.

To help customize your security policy and its attack objects to your network vulnerabilities, create custom groups of attack objects using the Object Editor. You can create static groups, which contain only the objects you specify, or dynamic groups, which contain objects based on matching criteria you specify.

Static Groups

A static group contains a specific, finite set of attack objects. Although the attack objects in the group can receive updates during the weekly signature update, no new attack objects are added.

Use static groups for the following tasks:

- To define a specific set of attacks to which you know your network is vulnerable
- To group your custom attack objects
- To define a specific set of informational attack objects that you use to keep you aware of what is happening on your network

Static groups require more maintenance than dynamic groups because you must manually add or remove attack objects in a static group to change the members. However, you can include a dynamic group within a static group to automatically update some attack objects.

Dynamic Groups

A dynamic group contains a dynamic set of attack objects that automatically updates its members during a weekly attack object update. The update adds or removes attack objects from each dynamic group based on the group criteria, eliminating the need to review each new signature to determine if you need to use it in your existing security policy.

Example: Creating Dynamic Groups

To create a dynamic group, you set the criteria for inclusion in the group using the following filters:

- Use a Service filter to add attack objects based on the attack service, such as FTP, HTTP, NetBios, and so on.
- Use a Products filter to add attack objects based on the application that is vulnerable to the attack.
- Use a Direction filter to add attack objects based on the direction of the attack traffic. You can select from the following directions:
  - *Any* monitors traffic from client to server and server and client.
  - *Client to Server* monitors traffic only from client to server (most attacks occur over client-to-server connections).
  - *Server to Client* monitors traffic only from server-to-client.
- Use a Last Modified (Until) filter to add attack objects that were modified on or before a selected date.
- Use a Last Modified (Since) filter to add attack objects that were modified on or since a selected date.
- Use a Severity filter to add attack objects based on the attack severity.

NOTE: All predefined attack objects are assigned a severity level by Juniper Networks. However, you can edit this setting to match the needs of your network.

- Use a False Positives filter to add attack objects based on the frequency that the attack produces a false positive on your network.
- Use a Category filter to add attack objects based on category.
- Use an Attack Type filter to add attack objects based on the type of attack object (signature or protocol anomaly).
- Use a Recommended filter to add attack objects based on whether Juniper Networks recommends them.

You create filters one at a time; as you add each criteria, IDP compares it to the predefined attributes for each attack object and immediately filters out any attack object that does not match. If you create a filter with attributes that no attack object can match, a message appears warning you that your dynamic group has no members.

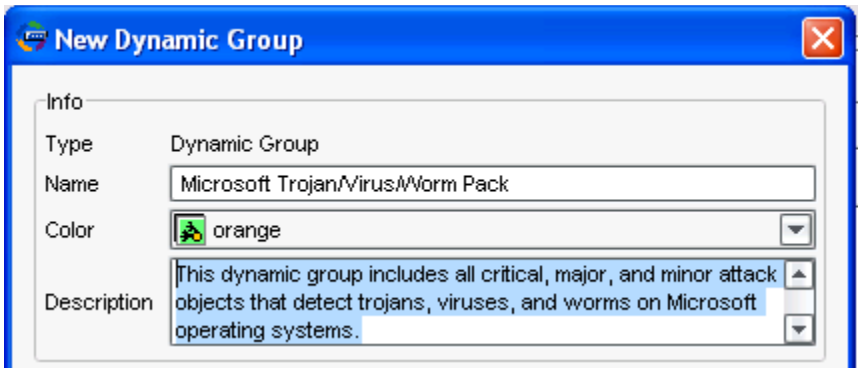
From the resulting list of matching attack objects, you can then exclude any attack objects that produces false positives on your network, or an attack object that detects an attack you are not vulnerable to.

NOTE: A dynamic group cannot contain another group, (predefined, static, or dynamic). However, you can include a dynamic group as a member of a static group.

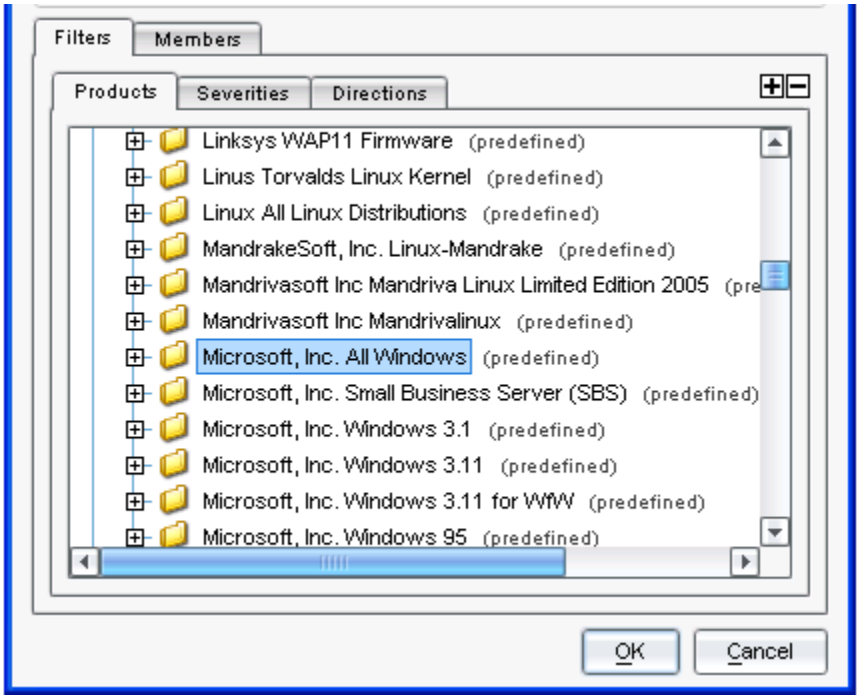
Example: Trojan Dynamic Group

To create a dynamic group:

1. From the navigation tree, select Object Manager > Attack Objects > IDP Objects.
2. Select the Custom Attack Group tab.
3. Click the + button and select Add Dynamic Group.
4. Enter a Name, Color, and Description for the group, as shown in the following figure.



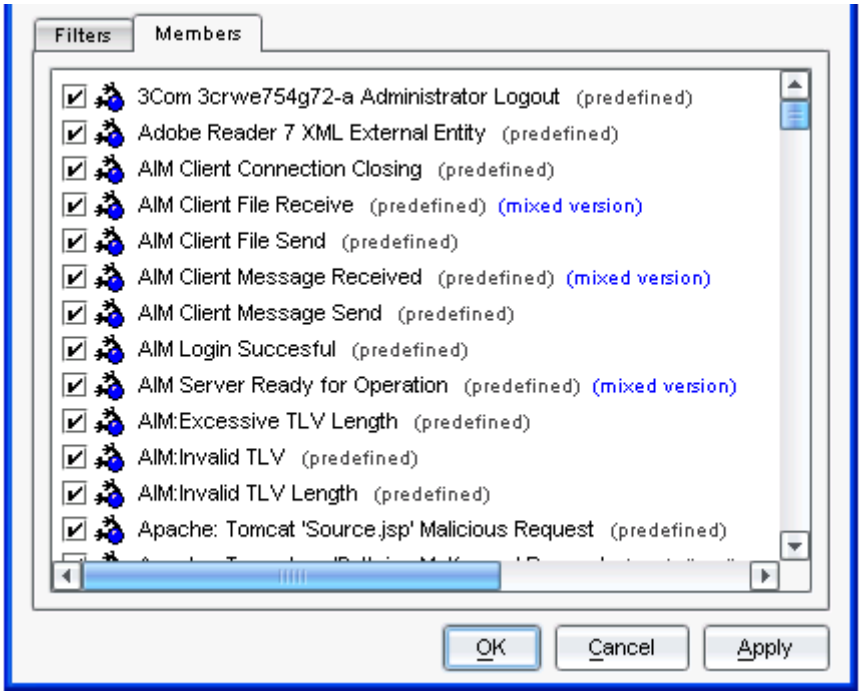
5. Next, add the filters that determine which attack objects should be in the group, as shown in the following figure.



- Most Trojans, viruses, and worms use client-to-server connections, so create a Direction filter to add attack objects based on the direction the attack traffic travels.
- You want to protect your Microsoft desktops and servers, so you should create a Products filter to add attack objects that detect attacks against any Microsoft operating system.
- Create a Severity filter to add attack objects that have a severity level of Critical, Major, or Minor.

IDP automatically applies all filters to the entire Attack Object database, identifies the attack objects that meet the defined criteria, and adds the matching objects as members of the group.

6. View the members of the group, as shown in the following figure.



7. To exclude an attack object from the list, deselect the checkbox next to the attack object.

8. Enter a comment about the filters, if desired, then choose a color to represent the dynamic group. Click OK to save the dynamic group.

Updating Dynamic Groups

When you are satisfied with the group criteria and its members, use the group in a security policy. The next time you update your attack objects, the update automatically performs the following steps:

- For all new attack objects, compares the predefined attributes of each attack object to each dynamic group criteria and adds the attack objects that match.
- For all updated attack objects, removes attack objects that no longer meet their dynamic group criteria. The update also reviews updated attack objects to determine if they now meet any other dynamic group criteria and adds them to those groups if necessary.
- For all deleted attack objects, removes the attack objects from their dynamic groups.

You can also edit a dynamic group manually, adding new filters or adjusting existing filters to get exactly the type of attack objects you want. However, you cannot edit a dynamic group from within a security policy; you must locate the group in the Object Editor to make changes to the dynamic group object.

Predefined Dynamic Groups

The IDP system includes several prebuilt dynamic groups for many operating systems and applications. You can use these prebuilt dynamic groups in your security policies, just as you would a custom dynamic group; the attack object update performs the same task for all dynamic groups.

Updating the Attack Object Database

Attackers are constantly devising new and better ways to infiltrate your network. The network-security community is constantly discovering these new attacks and creating new signature attack patterns to match them. To ensure that IDP remains highly effective against all emerging and evolving threats, Juniper Networks provides updates to the Attack Object database every work day and, if necessary, on an emergency basis. Updates can include:

- Additional attack objects
- Modification of descriptions or severities for existing attack objects
- Removal of obsolete attack objects

To update your Attack Object database, select Tools > View/Update NSM Attack Database. For more information about updating attack objects, refer to the *NetScreen-Security Manager Administrator's Guide.*

Searching Attack Objects

As you design your security policy, you might want to create a rule that detects a highly publicized attack (such as the NIMDA worm) or a rule that detects attacks against a specific product (such as Cisco routers).

To find a specific attack object or attack object group, you can search in any column using the following methods:

- Basic search: To find the string anywhere in the column, click on the column and start typing. Press the down arrow on your keyboard to find the next match.
- Advanced Search: To see more options, click on a column and press Ctrl-F. The following options appear:
  - C—Same as Basic search.
  - S—Restricts the search to entries that start with the string you type.
  - R—Searches using regular expressions.
  - I—Searches for an IP address. The wildcard character * can be used.

Displaying Attack Object Usage

To display all uses of an attack object or attack object group, right-click the object and select Find Usages.

Chapter 9

Configuring Sensor Settings

This chapter covers the following topics:

- Configuring Load-Time Parameters
- Configuring Router Parameters
- Configuring Run-Time Parameters
- Configuring Protocol Thresholds
- Device Templates

Sensor settings specify how the IDP Sensor handles traffic before it becomes associated with a connection. When you enable and start IDP on a device, default values for all IDP Sensor parameters are used. As you fine-tune a security policy to fit network traffic, you may want to edit these default values.

Sensor settings are associated with each Sensor in the Security Devices list.

The following sections describe groups of parameters.

Configuring Load-Time Parameters

To access Sensor settings, select Device Manager > Security Devices. Double-click the Sensor you want to work with. Select Sensor Settings from the Device dialog.

Use the Load Time Parameters tab to configure the following:

- Flow table size. For improved IDP performance, set the flow table size to limit the size of the connection table. This setting should reflect the maximum number of concurrent flows you expect to have at any one time. A TCP connection has about two flows per session, and a UDP connection has about three flows per session. *Default setting*: The IDP Sensor connection table can handle 100,000 concurrent flows. If you change this value, you have to restart the Sensor.

- Enable log suppression. Log suppression reduces the number of logs displayed in the Log Viewer by displaying a single record for multiple occurrences of the same event. Log suppression can negatively impact Sensor performance if the reporting interval is set too high. *Default setting*: Enabled. The IDP Sensor suppresses multiple instances of the same log record.

- Include destination IP's while performing log suppression. When log suppression is enabled, multiple occurrences of events with the same source, service, and matching attack object generate a single log record with a count of occurrences. If you enable this option, log suppression will only combine logs records for events with a matching source as well. *Default setting:* Not enabled. The IDP Sensor does not consider destination when determining matching events for log suppression.

- Number of log occurrences after which log suppression begins. This setting specifies how many instances of a specific event must occur before log suppression begins. When you look at the log records in the Log Viewer, you see this number of log records followed by a combined record for the remaining occurrences within the time interval. *Default setting:* 1 (log suppression begins with the first occurrence).

- Maximum number of logs that log suppression can operate on. When log suppression is enabled, IDP must cache log records so that it can identify when multiple occurrences of the same event occur. This setting specifies how many log records are tracked simultaneously by IDP. *Default setting:* IDP can operate on 16384 log records.

- Time (seconds) after which suppressed logs will be reported. When log suppression is enabled, IDP maintains a count of multiple occurrences of the same event. After the specified number of seconds has passed, IDP writes a single log entry containing the count of occurrences. *Default setting:* IDP reports suppressed logs after 10 seconds.

Configuring Router Parameters

Use the router parameters to control how the IDP Sensor handles Address Resolution Protocol (ARP) requests and replies and Media Access Control (MAC) address issues.

NOTE: The virtual router on the IDP Sensor is actually a virtual path, a logical grouping of the Sensor's network interfaces that creates a logical circuit for a network segment's traffic flow through IDP. The IDP virtual router does not support the same functions as a virtual router on a Juniper Networks firewall/VPN device.

To access Sensor settings, select Device Manager > Security Devices. Double-click the Sensor you want to work with. Select Sensor Settings from the Device dialog, then select Router Parameters.

- ARP timeout (seconds). Default: 3600 seconds. When the virtual router is in proxy-ARP mode, this setting controls how long an ARP entry is maintained in the virtual router. If the Sensor does not receive an ARP reply before the timeout expires, the ARP entry times out.

- ARP proxy timeout (seconds). Default: 20 seconds. In proxy-ARP mode, the IDP Sensor sends out proxy ARPs on all interfaces except the one on which an ARP request was received. This setting indicates how long the original ARP entry is maintained in the virtual router if the Sensor does not receive an ARP reply through that interface.

- Log ARP attacks. Default: Enabled. When selected, the IDP Sensor detects and logs all spoofed ARP requests/replies and other ARP anomalies.
- MAC timeout (seconds). Default: 3600 seconds. When the virtual router is in bridge mode, this setting controls how long a MAC entry is maintained in the virtual router.
- MAC proxy timeout (seconds). Default: 20 seconds. In bridge mode, the IDP Sensor performs MAC discovery if the target MAC address is not in its MAC table. This setting controls how long the entry is maintained in the virtual router until a reply comes back.

Configuring Run-Time Parameters

To access Sensor settings, select Device Manager > Security Devices. Double-click the Sensor you want to work with. Select Sensor Settings from the Device dialog.

Use the Run-Time Parameters tab to configure the following:

- Backdoor Detection. These settings control how the IDP Sensor implements heuristics. These parameters complement the rules in the backdoor rulebase.
  - Minimum interval between consecutive small packets (microseconds)—See Maximum interval description.
  - Maximum interval between consecutive small packets (microseconds)—Minimum and maximum intervals (in microseconds) between the arrival of two consecutive small packets in suspected interactive traffic. If the Sensor sees small packets arrive in less than the minimum or more than the maximum number of microseconds, it does not consider the traffic to be interactive.
  - Byte threshold for packet sizes in a backdoor connection (bytes)—This setting controls the maximum number of bytes a TCP packet must contains before the Sensor uses the packet for backdoor detection heuristics.
  - Minimum number of data carrying TCP packets—This setting controls the minimum number of data-carrying TCP packets in suspected interactive traffic.
  - Minimum percentage of back-to-back small packets (percentage)—This setting controls the minimum percentage of consecutive small packets in a suspected interactive traffic. If the Sensor sees less than this percentage, it does not report a backdoor event.
  - Ratio of small packets to the total packets (percentage)—This setting controls the minimum percentage of small packets that the Sensor uses for backdoor detection heuristics. If the Sensor sees less than this minimum, it does not report a backdoor event.
- Flow Management. These settings control how the IDP Sensor handles flows. Each connection typically has two flows, one in each direction.

- Timeout (seconds) for non-UDP/TCP/ICMP flows—Each connection through the IDP typically has two flows, one in each direction. If the Sensor does not see flow activity for the specified timeout, it removes the idle flow from the flow table.
- Timeout (seconds) for UDP flows—Each connection through the IDP typically has two flows, one in each direction. If the Sensor does not see flow activity for the specified timeout, it removes the idle flow from the flow table.
- Timeout (seconds) for TCP flows—Each connection through the IDP typically has two flows, one in each direction. If the Sensor does not see flow activity for the specified timeout, it removes the idle flow from the flow table.
- Timeout (seconds) for ICMP flows—Each connection through the IDP typically has two flows, one in each direction. If the Sensor does not see flow activity for the specified timeout, it removes the idle flow from the flow table.
- Maximum TCP sessions—Controls the maximum number of sessions that IDP maintains. If the Sensor reaches the maximum, it drops all new sessions and sends a SESSION_LIMIT_EXCEEDED log to the Management Server.
- Maximum UDP sessions—Controls the maximum number of sessions that IDP maintains. If the Sensor reaches the maximum, it drops all new sessions and sends a SESSION_LIMIT_EXCEEDED log to the Management Server.
- Maximum ICMP sessions—Controls the maximum number of sessions that IDP maintains. If the Sensor reaches the maximum, it drops all new sessions and sends a SESSION_LIMIT_EXCEEDED log to the Management Server.
- Maximum IP (non-TCP/UDP/ICMP) sessions—Controls the maximum number of sessions that IDP maintains. If the Sensor reaches the maximum, it drops all new sessions and sends a SESSION_LIMIT_EXCEEDED log to the Management Server.
- Reset flow table with policy load/unload—IDP keeps track of connections in a table. If selected, the Sensor resets the flow table each time a Security Policy loads or unloads. If this setting is unselected, then the Sensor continues to remember a previous Security Policy until all flows referencing that Security Policy go away. Juniper Networks recommends that you keep this setting enabled to preserve memory.
- Log flow related errors—Writes flow-related errors to the log. A flow-related error is when IDP receives a packet that doesn't fit into expected flow. Examples:
  - Receipt of a packet that does not match any known session or flow
  - Receipt of a packet for a session after the session threshold has been reached

- Intrusion Detection. These settings control how the IDP Sensor handles intrusion detection settings.
  - Buffer Overflow Emulator—Turns on buffer overflow emulation.
  - Attack matches per packet when Signature Hierarchy(0 to disable) take effect—Sets the threshold for activating Signature Hierarchy calculations.

    Common attack can be composed of several known vulnerabilities. Each vulnerability has an attack object, and each would generate a separate log entry if the Signature Hierarchy feature is disabled.

    Example: HTTP:IIS:COMMAND-EXEC. For a policy with Critical, High, Medium, Low & Info attacks and logging enabled, a single detection of HTTP:IIS:COMMAND-EXEC attack generates the following logs.
    - HTTP:IIS:COMMAND-EXEC [wininnt/system32/cmd.exe] (medium)
    - HTTP:WIN-CMD:WIN-CMD-EXE [cmd.exe] (medium)
    - HTTP:REQERR:REQ-MALFORMED-URL [anomaly for %xx] (medium)
    - HTTP:DIR:TRAVERSE-DIRECTORY (anomaly for ../) (medium)
    - HTTP:REQERR:REQ-LONG-UTF8CODE (anomaly for œ) (medium)
    - TCP:AUDIT:BAD-SYN-NONSYN (info)
    - HTTP:AUDIT:URL (info)
    - TCP:AUDIT:BAD-SYN-NONSYN (info)
    - HTTP:STC:SRVRSP:404-NOT-FOUND (info)

    If the number of attacks in a packet exceeds the set value, then the Sensor examines its signature hierarchy to see if some attacks are actually part of a larger attack. If so, then only the parent attack is displayed in the logs. In this example, if the value was set to 9 or lower, then only a log for HTTP:IIS:COMMAND-EXEC would be generated.

    An attack in the signature hierarchy may have multiple parents or multiple children. If a child attack is part of two discovered parents, the Sensor takes action based on the parent with the highest severity.

- IP Actions. These settings control how the IDP Sensor handles IP actions. These parameters relate to the IP actions used in rules.
  - Reset block table with policy load/unload—The block table maintains the state of active IP actions. The Sensor can reset the block table each time a Security Policy loads or unloads.

- Run-time parameters. These settings control how the IDP Sensor handles Remote Procedure Call (RPC) requests and replies and IP fragments. Click the Show button to view and configure these settings.

- RPC timeout (seconds)—The Sensor performs a stateful inspection of all RPC messages on port 111, then builds a table of program to port mapping for each RPC server that it finds on the network. This setting indicates how long an entry in the table is maintained.
- RPC transaction timeout (seconds)—All RPC messages (port 111) are based on a request/response protocol. When the Sensor receives a request, it adds the request to a request table. If the Sensor does not receive an RPC reply in the specified timeout, the RPC entry times out.
- Exempt management server flows—When the Sensor and the IDP Management Server are run on different machines, selecting this setting exempts all IDP Management Server connections from going through the normal packet processing mechanisms. This means that the Sensor does not match rules from the various rulebases for these connections.
- Fragment timeout (seconds)—Controls when the Sensor drops an incomplete fragment chain because one or more fragments did not arrive. If the Sensor does not receive missing fragments in the specified timeout, it sends a FRAGMENT_TIME_EXCEEDED log record to the IDP Management Server.
- Minimum fragment size (bytes)—When set to a non-zero value, the Sensor drops all IP fragments less than the specified size in bytes.
- Maximum fragments per IP datagram (bytes)—An IP datagram can be broken into many fragments which, when assembled, should not exceed 64 K. Since IP fragment processing is CPU and memory intensive, this setting controls the size of the IP fragment chain. If the number of fragments in a chain exceeds this number, then the Sensor drops the entire fragment chain.
- Maximum concurrent fragments in queue—The Sensor can perform pseudo reassembly of IP fragment chains. This setting controls the maximum number of reassembled fragment chains. Once this limit is reached, the Sensor drops all new IP fragment chains and sends a TOO_MANY_FRAGMENTS log to NSM. If your network produces a large number of IP fragments, such as those produced by Network File System (NFS), increase the number of fragments per chain to eliminate unnecessary logs.
- Log fragment related errors—Select if you want the Sensor to log fragment related errors.
- Enable GRE decapsulation support—Turns on GRE decapsulation. Other GRE decapsulation settings are controlled from the Sensor CLI. See “GRE Decapsulation Constants” on page 179 for more information.
- Enable GTP decapsulation support—Turns on GTP decapsulation. Other GTP decapsulation settings are controlled from the Sensor CLI. IDP supports UDP GTPv0 and GTPv1 only. See “GTP Decapsulation Constants” on page 180 for more information.

- Enable SSL decapsulation support—Turns on SSL decapsulation. Other SSL decapsulation settings are controlled from the Sensor CLI. See "SSL Constants" on page 180 for more information.

- SYN-Protector. These settings control how the IDP Sensor handles SYN flows. These parameters complement the rules in the SYN-Protector Rulebase. See "SYN-Protector Constants" on page 181 for more information.

  - Timeout for half-open SYN protected flows—Passive mode only. The number of seconds the Sensor will hold an incomplete SYN-ACK handshake before purging it.

  - Lower SYN's-per-second threshold below which SYN protector will be deactivated—IDP 4.0 uses SYN Cookies to protect from SYN-flood attacks. If the number of SYN packets to a destination exceeds this number per second, the Sensor will begin adding SYN Cookies to its SYN-ACK packets. If the number of SYN packets per second to the destination falls below this number, the Sensor will stop adding SYN Cookies. This setting has a slightly different function in Passive mode, described in the next setting.

  - Upper SYN's-per-second threshold above which SYN protector will be activated—As of IDP 4.0, this setting is only meaningful for Passive mode. In passive SYN-Protector mode, if the number of SYN packets per second received by the Sensor to the destination exceeds the lower threshold *plus* the upper threshold, the Sensor begins treating SYNs and a SYN-flood. The Sensor continues to treat the SYNs as a SYN-flood until the number of SYN packets per second falls below the lower threshold.

- TCP Reassembler. These settings control how the IDP Sensor handles TCP flows. Click the Show button to view and configure these settings.

  - Ignore packets in TCP flows where a SYN hasn't been seen—The absence of SYN flags in TCP flows is suspect, yet still a very common occurrence. The Sensor can ignore packets within TCP flows that do not yet contain a SYN flag.

  - Close flows as soon as a FIN is seen (recommended)—When a TCP connection closes, the Sensor sees a FIN packet from each side of the connection followed by an ACK packet from each side of the connection. However, TCP does not guarantee delivery of the final ACK. This setting enables the Sensor to quickly close a TCP connection after receiving a FIN packet. When enabled, the Sensor maintains a connection waiting for a final ACK for 5 seconds, then closes the connection.

  - Timeout for connected, idle TCP flows (seconds)—This setting controls the number of seconds that the Sensor maintains connected (but idle) TCP flows.

  - Timeout for closed TCP flows (seconds)—When the Sensor sees a RST packet or FIN/FIN + ACK packets on a TCP connection, it closes the connection flows. The Sensor drops any further packets for the closed flow, but does not delete existing, closed flows from the flow table. This setting controls the number of seconds that closed TCP flows are maintained in the flow table.

- Timeout for CLOSE-WAIT/LAST-ACK TCP flows (seconds)—When a TCP connection closes, the Sensor sees a FIN packet from each side of the connection followed by an ACK packet from each side of the connection. However, TCP does not guarantee delivery of the final ACK. This setting controls the number of seconds a connection is maintained while waiting for the final ACK. To improve Sensor performance during heavy loads, decrease the timeout—this reduces the size of the flow table by closing connections sooner.

- Traffic Signatures. These settings control how the IDP Sensor handles scan-related issues. These parameters complement the rules in the Traffic Anomalies rulebase.

  - Byte threshold for suspicious flows—A scan typically uses small packets to access its targets. You can exclude suspicious flows that contain large packets to prevent false positives when detecting scans. If the Sensor sees more than this maximum, it does not consider the connection to be a scan.

  - Reporting frequency when scan is in progress (seconds)—Attackers can perform blatant scans very quickly, mapping your network in just a few seconds, but these scans typically trigger IDSes and leave evidence behind. Stealthy scans are performed over much longer time periods, lasting hours, days, or even weeks, making them more difficult to detect. This setting controls how often the Sensor generates "in progress" logs for a stealthy scan.

  - The number of IP tracked for session rate—This setting controls the number of IP addresses tracked by the session rate counter. If the Sensor sees more addresses than the maximum, it does not track the additional IP addresses.

Configuring Protocol Thresholds

To access Sensor settings, select Device Manager > Security Devices. Double-click the Sensor you want to work with. Select Sensor Settings from the Device dialog.

Use the Protocol Thresholds and Configuration tab to control how the IDP Sensor handles packets for specific types of protocols. Click the Show button to view and configure settings for a specific protocol. For detailed descriptions of each setting, refer to the NSM *Online Help*.

Device Templates

All of the previously described settings can also be configured in a device template, which can be pushed to multiple Sensors. Sensor settings are found in the Security > IDP SM Settings section of a device template.

Refer to the *NetScreen-Security Manager Administrator's Guide* for more information about device templates.

Chapter 10

Using Tagged Interfaces and Virtual Routers

This chapter covers the following topics:

- Using Tagged Interfaces (802.1Q)
- Using Virtual Routers

A virtual local area network (VLAN) is an emulation of a local area network (LAN), a group of network components that communicate using a subnet of a network. A VLAN, however, enables you to further subdivide network addressing space into a virtual subnet—like a subnet within a subnet. Each network component that is a member of a VLAN has a *tagged* interface mapped onto a physical interface of the component. This tagged interface is a virtual interface that handles all traffic to and from the VLAN but that ignores all other traffic that passes through the physical interface. The individual packets in the VLAN traffic carry a tag header in their Ethernet frame that identifies the VLAN they belong to, enabling the network switch to forward packets to the correct destination.

A virtual router is an emulation of a physical router, running on a physical device such as a router or the IDP Sensor. Virtual routers can group physical and virtual interfaces together, limiting the forwarding options for incoming packets to the other interfaces in the group.

NOTE: The virtual router on the Sensor is actually a virtual path, a logical grouping of the Sensor's network interfaces creating a logical circuit for a network segment's traffic traveling through IDP. The IDP virtual path does not support the same functionality as a virtual router on a Juniper Networks firewall/VPN device.

Using 802.1Q (the VLAN protocol) and tagged interfaces, you can connect a single physical IDP Sensor interface to multiple VLANs on a network switch. Then, use virtual routers to define the IDP interfaces (physical and tagged) that can communicate with each other. Interfaces not attached to the same virtual router cannot forward traffic to one another.

Using Tagged Interfaces (802.1Q)

The IDP Sensor supports the 802.1Q VLAN protocol at the kernel level and can generate 802.1Q frames in all in-line IDP deployment modes. The Sensor receives tagged and untagged VLAN traffic through multiple virtual interfaces and processes packets according to the installed security policy:

- Sniffer mode. IDP removes VLAN tags from incoming frames and processes the packet.
- Transparent mode. IDP reads the VLAN tags, but passes them through unaltered.
- Other in-line modes. IDP removes VLAN tags from incoming frames, processes the packet, and retags outgoing frames as necessary. For untagged (physical) interfaces, IDP generates normal Ethernet frames for all packets. For tagged virtual interfaces, IDP generates the appropriate 802.1Q tag for all packets.

A virtual interface uses the following format:

eth<physical interface>.<tag header>

For example, eth2.10 indicates that eth2 has a tagged VLAN interface of 10. A physical interface can contain multiple virtual interfaces.

NOTE: You do not need to configure VLAN tags for interfaces on a Sensor in Transparent mode. The Sensor automatically forwards traffic correctly, whether or not it is tagged for a specific VLAN.

Forwarding Traffic Through the IDP Sensor

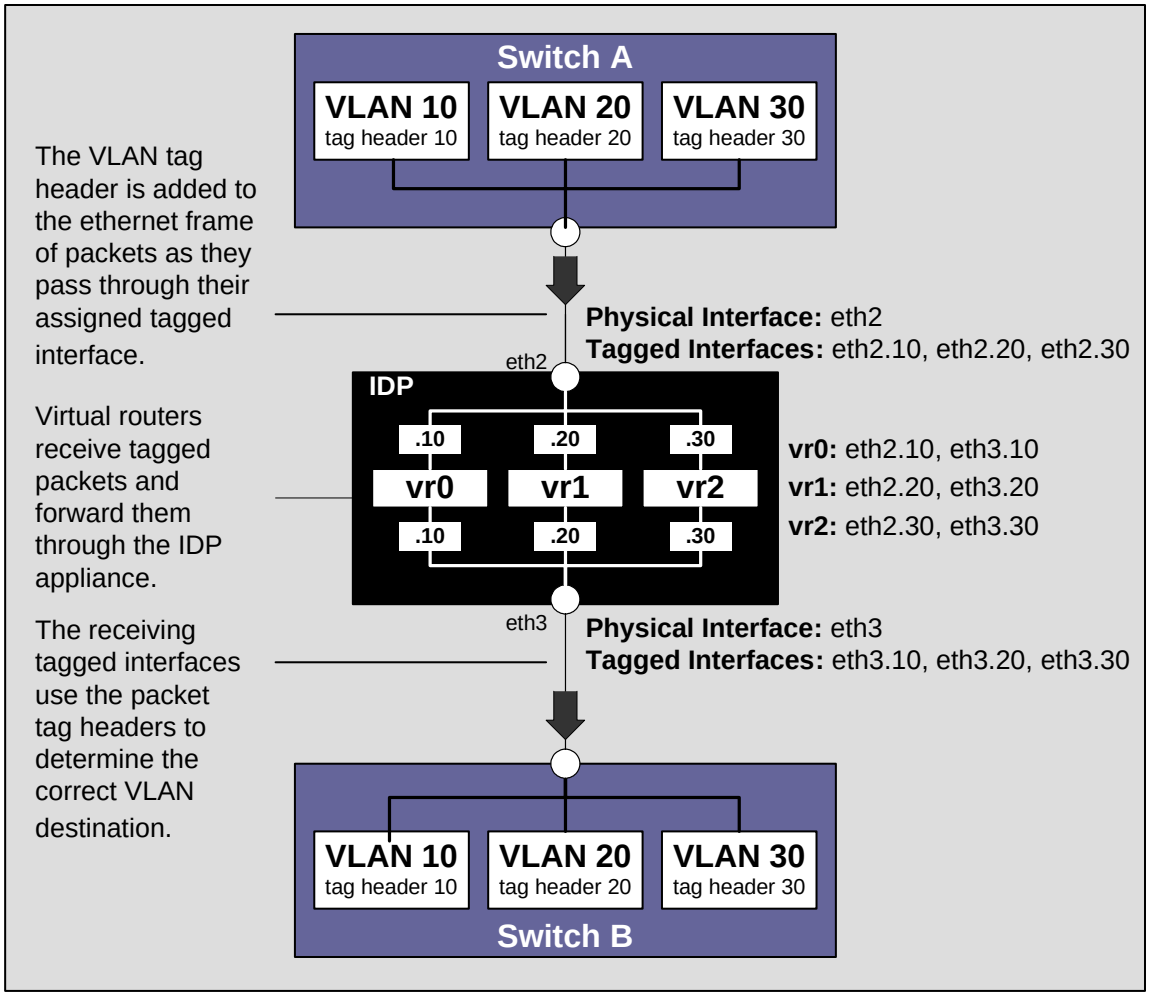
You must use virtual routers to restrict the interfaces (tagged or untagged) that can forward traffic to each other. Each virtual router creates a forwarding interface group; only interfaces that belong to the same virtual router can forward traffic to each other. An interface can belong to only one virtual router.

- Untagged interfaces. If you are not using virtual interfaces to tag packets, you can forward traffic through IDP using multiple virtual routers attached to separate physical interfaces. The Sensor processes all incoming packets from a physical interface according to the installed security policy, but maintains separate ARP/MAC tables for each virtual router. To transmit processed packets, the Sensor uses the virtual-router ARP/MAC table to determine the next hop and forwards packets with a normal Ethernet frame (without the tag header). The forwarding interfaces on a Sensor in Transparent mode must be assigned to unique virtual routers in pairs—traffic coming in one interface is forwarded out the other interface in the virtual router. In Transparent mode, the Sensor forwards traffic with the appropriate VLAN tag even though the Sensor interface is not itself assigned a tag.
- Tagged interfaces. If you are using virtual interfaces to tag packets, you can forward traffic through IDP using multiple virtual routers attached to the same physical interface. The Sensor processes all incoming packets from each virtual interface according to the installed security policy and maintains separate

ARP/MAC tables for each virtual router. To transmit processed packets, the Sensor uses the virtual-router ARP/MAC table to determine the next hop and forwards packets with the tag header attached to the Ethernet frame (an 802.1Q frame).

- Tagged and untagged interfaces. You can use both virtual and physical interfaces to forward traffic through the IDP Sensor. For outbound packets, the Sensor uses tagged frames for packets that use a virtual interface and normal Ethernet frames for packets that use a physical interface.

Figure 21: Forwarding Tagged Traffic Through the IDP Sensor



Configuring VLANs with the ACM

When you configure your IDP Sensor using the Appliance Configuration Manager (ACM), you can enable VLAN support on the IDP appliance and configure tagged interfaces:

- You can use the management interface on the IDP Sensor as a tagged interface.
- You cannot use the state-sync (HA) interface as a tagged interface.

- You cannot pass untagged traffic through a tagged interface, except with a Sensor configured in Transparent mode.
- You can pass traffic for up to 64 VLANs through the IDP 100, 200, 500, 600, 1000, and 1100 appliances. You can pass traffic for up to 4 VLANS through the IDP 10 and 50 appliances.

When using virtual and physical interfaces, be sure to configure the IDP Sensor to avoid IP address collision within multiple VLANs. For step-by-step instructions on configuring VLANs for the IDP appliance, refer to the ACM *Online Help*.

Working with Untagged Root Sys VLANs

If your network includes a Juniper Networks firewall appliance NS-200 or higher that uses the untagged root sys VLAN, you must tag the VLAN traffic before it can pass through the IDP appliance using one of the following options:

- You can reconfigure your firewall to tag the root sys VLAN traffic.
- You can configure a switch positioned between the firewall and IDP to receive incoming root sys VLAN traffic and tag it before forwarding to the IDP appliance.

NOTE: This procedure is not necessary if your Sensor is configured in Transparent mode.

Command Line Interface Options

Use the sctop and scio command line utilities to get information on virtual routers and VLANS.

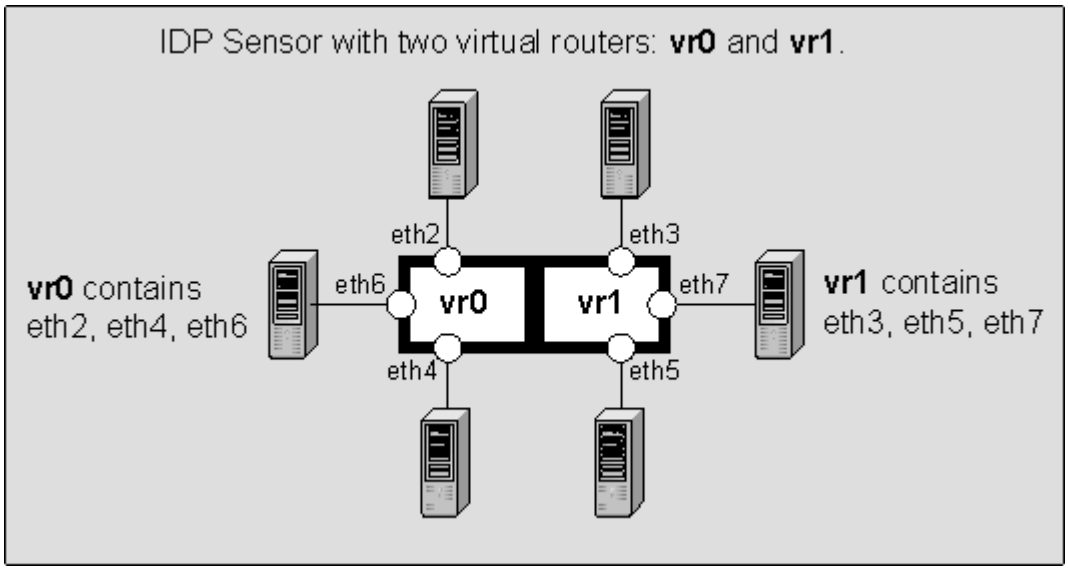
- -v to specify the name of the virtual router
- -v < name of vr > to view virtual router flows.
- scio vr to list virtual routers. Use to monitor each virtual-router ARP/MAC table independently.
- < clist virtual circuits > lists virtual interface and VLAN information.

Using Virtual Routers

To control how IDP passes traffic, create a virtual router that groups two or more interfaces (physical or virtual) on the IDP Sensor; incoming traffic from one interface is sent to another interface in the virtual-router group.

You can configure multiple virtual routers on an IDP Sensor, as shown in Figure 22.

Figure 22: Multiple Virtual Routers on IDP Sensor



The default virtual router is vr0; subsequent virtual routers are user-defined. You can configure and manage each virtual router as an independent entity.

Configuring Virtual Routers with the Appliance Configuration Manager

When you configure your IDP Sensor using the Appliance Configuration Manager (ACM), you can enable virtual-router support on the IDP Sensor. You can use multiple virtual routers to connect virtual interfaces on the IDP Sensors; however, you must use the same deployment mode for each virtual router.

For step-by-step instructions on configuring virtual routers for the IDP Sensor, refer to the ACM *Online Help*.

Chapter 11

Implementing High Availability

This chapter covers the following topics:

- Standalone High Availability
- Implementing Standalone High Availability
- Standalone HA Switch Compatibility
- Switch Hardware and Configuration
- External High Availability
- Spanning Tree Protocol

In high availability (HA) mode, multiple IDP Sensors combine into a cluster. An HA solution can provide your network with increased throughput using load sharing and increased reliability using failure protection.

IDP provides the following HA solutions for your network:

- Standalone HA solution. Requires two or more IDP Sensors running in active-gateway Router or active-gateway Proxy-ARP mode and minimal network reconfiguration.
- External HA solution. Requires two or more IDP Sensors running in active-gateway Bridge, active-gateway Transparent, or active-gateway Router mode, and two external HA devices, such as firewalls or load balancers, that can provide failure protection and/or load balancing.

You can also use Spanning Tree Protocol (STP) to provide failover for IDP Sensors running in Bridge or Transparent mode. For details, see “Spanning Tree Protocol” on page 174.

NOTE: IDP Sensors in Transparent mode do not actively participate in STP, but they do pass the BPDUs used by STP.

Standalone High Availability

The standalone high availability (HA) solution combines two IDP Sensors in Proxy-ARP mode or Router mode. When you join two IDP Sensors, you create a single virtual IDP, known as an *HA cluster*. Network devices such as switches, routers, and servers handle the HA cluster as a single device when forwarding and receiving traffic. Such devices are unaware of how many Sensors exist in the HA cluster or of their individual status.

The IDP Sensors in an HA cluster are known as *nodes*; both nodes in the HA cluster receive traffic from the forwarding switch, but only one node actually handles the packet. All nodes are aware of the status of other nodes in the HA cluster, so if one or more nodes fails, the remaining nodes immediately begin handling extra traffic.

The IDP system supports standalone HA configurations in Proxy-ARP and Router modes. To use an HA solution, you must have the following:

- Identical model and configuration for all IDP Sensors.
- Two IDP Sensors running the same version of IDP Sensor software.
- A minimum of three interfaces per IDP Sensor (two forwarding, one HA).
- NetScreen-Security Manager.
- Network switches that support IGMP *and* pass Layer 2 multicast *or* unicast to multiple ports. If Layer 2 multicast is chosen, your network devices must learn or be configured to learn multicast ARP.
- An available IP address for each forwarding interface on each IDP Sensor.
- A single entry point for network traffic on each side of the IDP Sensor.

NOTE: The IDP Sensor does not support cross-bar configurations that use redundant forwarding mechanisms for a single interface.

Traffic Handling

Typically, traffic enters your network by passing through a network switch. In an IDP HA solution, you configure your network switch to forward traffic to the HA cluster, where each node in the cluster receives a copy of all network traffic. Then, each node performs a simple algorithm on the traffic's source and destination addresses and determines whether to handle the traffic.

If the node determines that it should handle the traffic, it processes the data packets in that traffic according to the installed security policy. After processing the traffic, the node forwards allowed traffic to the destination and drops or blocks disallowed traffic. However, if the node determines that it should *not* handle the traffic, it ignores the traffic entirely. Each node makes the decision to handle or not handle traffic based on:

- The status of other nodes in the cluster (the node might need to handle traffic for a failed node).
- The status of the node itself (the node must be functioning normally to process traffic).
- The source and destination address of the traffic (the node handles a specific traffic connection, not random flows).

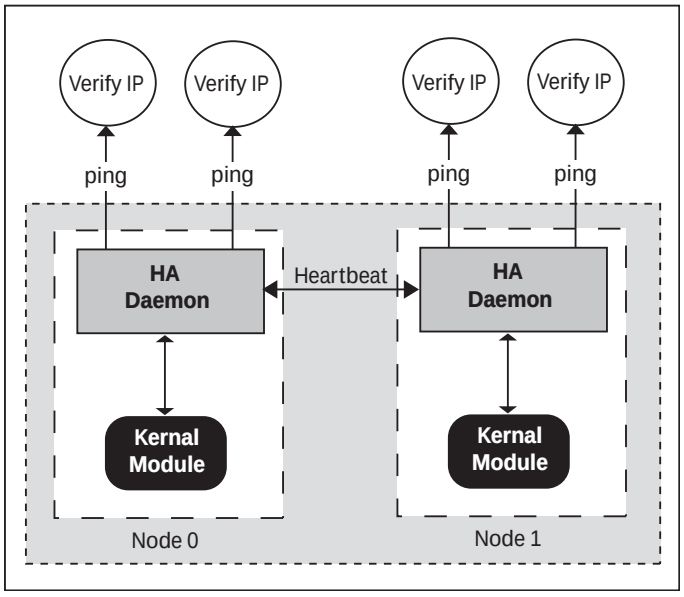
Node and Cluster Communications

A node obtains information about its own status and the status of the cluster from the HA daemon (a process that runs on each node in the cluster). A Sensor has its own HA daemon that communicates with other HA daemons on other Sensors. The HA daemons first identify the status of their own node, then exchange that information with other HA daemons, enabling each daemon to calculate the status of the cluster.

To determine the status of its node, the HA daemon performs the following tasks:

- Sends a ping request to a user-selected IP address to verify network connectivity and ensure that the Sensor can receive and forward traffic
- Uses the Heartbeat protocol to verify Sensor status and ensure that the Sensor is functioning normally

Figure 23: HA Daemon Operation



The daemon performs each function once per interval (a user-defined value that is measured in seconds).

Path Verification

The HA daemon uses ICMP to determine which interfaces on a node have connectivity to the rest of the network; this is known as path verification. To perform path verification, the daemon sends ICMP echo-requests (pings) to the IP address, called the Verify IP, of another device that is reachable by one of the node's forwarding interfaces.

The daemon pings the Verify IP once per interval, which is a user-defined time period (default is 1 second), and keeps track of every time it does not receive a response. If the daemon does not receive a response for a user-defined number of intervals (known as the fail-count), it considers the state of its node to be FAIL. (For information about node states, see "Determining Node Status" on page 146).

NOTE: In Proxy-ARP mode, each virtual router must be on a separate subnet or the Verify IP will fail. If all the virtual routers must be on the same subnet, then you must set routes for each interface to the Verify IP.

Example: HA Clusters

HA cluster 1 has an interval setting of 1 second and a fail-count of 3 seconds. Node A sends a ping to its Verify IP address for two consecutive intervals (2 seconds) but receives no response. If Node A sends another ping during the third interval (three intervals = fail-count) and still receives no response, Node A considers itself in FAIL state.

Interval and fail-count cluster settings must be the same for all nodes in the HA cluster.

Sensor Status (Heartbeats)

Heartbeats enable each daemon on each node to determine the state of the other nodes in the HA cluster. The HA daemons on each node send heartbeats on one or more forwarding interfaces. To reduce network and CPU overhead, heartbeats use IP Multicast.

You configure the heartbeat interface by giving it a heartbeat multicast IP address; all nodes on a subnet must use the same interface for the heartbeat interface; each heartbeat interface must also use the same multicast IP address. Per RFC 2365, Juniper Networks recommends that you use multicast IP addresses from 239.0.0.1 to 239.255.255.254 for the heartbeat interfaces.

Heartbeat networks can run through a hub or multicast-aware switches. Switches that have IGMP enabled can learn the Heartbeat IP multicast. If IGMP is not enabled on the network switch that connects the Sensors in the HA cluster, you must manually enable IGMP by editing the switch configuration file.

Switches that do not support IGMP cannot be used for a standalone HA configuration and are not supported.

To view the heartbeat packet data in tcpdump, use the Juniper Networks version of tcpdump. For more information, see "Heartbeats" on page 149.

Heartbeat Information and Processing

HA daemons send and receive heartbeat packets from other HA daemons. Each heartbeat packet contains information that describes the current state of the node it came from, as well as cluster-setting data, which is the same for all heartbeats.

When the HA daemon receives the heartbeat, it performs a series of checks to verify the authenticity of the heartbeat packet, then determines how to process the heartbeat information, as shown in Table 15.

Table 15: Heartbeat Processing

Check	Requirements
Packet Authentication	HB versions match. Heartbeat belongs to same cluster. Heartbeat timestamp is increased.
Configuration parameters match	Node ID is valid. Cluster settings match. Cluster IP and cluster MAC addresses match. Same number of interfaces with IP addresses.
Status of Remote Node	

If all checks succeed, the heartbeat packet is processed. If *any* check does not succeed the heartbeat is not processed.

Heartbeat Failure

The HA daemon keeps track of every time it does not receive a heartbeat from a specific node during an interval or it does not process a heartbeat because of a heartbeat-check failure. If the HA daemon does not receive or process a heartbeat from a specific node for a user-defined number of intervals (known as the fail-count), the daemon considers the state of the node to be FAIL.

Example: Heartbeats

HA cluster 1 has an interval setting of one second and a fail-count of three seconds. Node A does not send heartbeats to Node B or Node C for two intervals (two seconds). If Node A does not send a heartbeat by the third interval (three intervals = fail-count), Node B and Node C consider the state of Node A to be FAIL.

For information on node states, see “Determining Node Status” on page 146.

NOTE: Interval and fail-count are cluster settings that are the same for all nodes in the HA cluster.

Determining Node Status

The HA daemon determines which nodes in the HA cluster are capable of handling traffic. The HA daemon determines the state of its own node and other nodes in the cluster to make HA-cluster calculations.

- Local State Calculations. The HA daemon uses the information gathered from path verification and incoming heartbeat packets to determine the state (FAIL, INIT, or OK) of its local node, as shown in Table 16.

Table 16: Local Node State

During	The HA Daemon...	Moves To and From...
Fail	■ Causes the node it is on not to forward traffic, but sends heartbeats to other nodes informing them that it is in FAIL state. ■ Pings all Verify IPs (path verification). ■ Learns about network traffic via the state-sync protocol and updates the local state table. ■ Receives heartbeats from other nodes.	■ A node moves to the FAIL state from the INIT or OK state when path verification reaches the fail-count. ■ A node moves from the FAIL state to the OK state when path verification reaches the reintegrate-count (the number of intervals a path verification must succeed before OK).
Init	■ Does not forward traffic, but sends heartbeats to other nodes informing them that it is in INIT state. ■ Receives heartbeats from other nodes. ■ Pings all Verify IPs (path verification). ■ Learns about network traffic via the state-sync. ■ Updates its state table.	■ A node starts in the INIT state when the HA daemon starts or when it receives a HUP signal. ■ A node moves from the INIT state to the OK state when path verification reaches the reintegrate-count (the number of intervals a path verification must succeed before OK). ■ A node moves from the INIT state to the FAIL state when path verification reaches the fail-count. ■ If a node in the INIT state receives a heartbeat from a node in the OK state indicating that it has a different cluster configuration, the node in INIT state goes offline and the HA daemon exits.
OK	■ Forwards traffic and sends heartbeats to other nodes informing them that it is in the OK state. ■ Receives heartbeats from other nodes. ■ Pings all Verify IPs (path verification). ■ Learns about network traffic via the state-sync. ■ Updates its state table.	■ A node moves to the OK state from the FAIL or INIT state when path verification reaches the reintegrate-count (the number of intervals a path verification must succeed before OK). ■ A node moves from the OK to the FAIL state when path verification reaches the fail-count.

- Remote State Calculations. The HA daemon uses the information gathered from incoming heartbeat packets to determine the state (FAIL or OK) of all other nodes in the cluster.

- Making HA Cluster Calculations. The HA daemon uses the information gathered from the local and remote calculations to determine which nodes in the cluster are capable of handling traffic. This cluster information is then passed to the node's kernel module, which decides if the node should process the packet or ignore it.

HA Communication Settings

Nodes in an HA cluster communicate with each other (via the HA daemon) using the heartbeat interface on the IDP Sensor. When you configure the Sensor software on the IDP Sensor (using the ACM), you configure three types of communication settings for the HA cluster:

- Cluster settings must be the same for all nodes.
- Interface settings are the same for a given interface on all nodes.
- Node settings are different for each node.

These settings determine how the HA daemon verifies and manages the cluster state.

Cluster Settings

All nodes must use the same settings for HA daemon communication. These cluster settings determine the information that is included in the heartbeat packet, which enables the HA daemon to verify and manage the state of the cluster. A list of cluster settings, supported values, and descriptions is shown in Table 17.

Table 17: Cluster Settings

Setting	Value	Description	ACM Section
enable	yes, no	Enables or disables the HA daemon.	Configure High Availability
mode	lb (load balancing) hs (hot standby)	Determines the failover mode for the HA cluster.	Configure High Availability
Cluster ID	numeric, 0-5	Sets the HA cluster ID. Used to distinguish multiple IDP clusters.	Configure Standalone HA
Total Nodes	numeric, 2	Sets the total number of nodes (Sensors) used in the HA cluster.	Configure Standalone HA
Test/Heartbeat Interval	numeric, 1-10	Sets the number of seconds between heartbeats, Verify IP pings, and state calculations.	Configure Standalone HA
Failure Count	numeric, 1-10	Sets the number of intervals (using the interval setting): ■ A ping must fail before a node considers itself failed. ■ A heartbeat must not be received before the other node is considered failed.	Configure Standalone HA
Reintegrate Count	numeric, 1-10	Sets the number of intervals (using the interval setting) that all Verify IP pings must succeed before a node changes from: ■ INIT to OK. ■ FAIL to OK.	Configure Standalone HA
Shared Secret	alphanumeric, 6-16	Sets the string that is used to authenticate HA messages.	Configure Standalone HA

Interface Settings

All nodes in the HA cluster use identical interface settings for a given interface. For example, all eth2 interfaces on all Sensors in the HA cluster use the same interface settings. A list of interface settings, supported values, and descriptions is shown in Table 18.

Table 18: Interface Settings

Setting	Description	ACM Section
Cluster IP	The unicast IP address that is used to forward traffic	Configure Forwarding Cluster MAC/IP
Cluster MAC	The multicast or unicast MAC address that is used to forward traffic	Configure Forwarding Cluster MAC/IP
Verify IP	The IP address that the node pings to verify that the interface is up	Configure Link-Check
Heartbeat IP	The multicast IP address that is used to send heartbeats to other nodes	Configure Heartbeat
State-Sync	The dedicated IP address that is used to share state table information with other nodes	Configure HA State-Sync

Supported Modes

The IDP system supports standalone HA configurations in Proxy-ARP and Router modes:

- In Proxy-ARP mode, all IDP Sensors use the same multicast cluster MAC address for each side of the HA cluster.
- In Router mode, all IDP Sensors use the same unicast cluster IP addresses and multicast cluster MAC addresses for each side of the HA cluster.

Choosing a Forwarding Option

The standalone HA solution can use one of two Layer 2 forwarding options, unicast or multicast, as shown in Table 19.

Table 19: Forwarding Options

	Advantages	Disadvantages	Other Notes
Unicast MAC	Minimizes Layer 3 device problems.	Not supported by many switch vendors. Must configure switches that directly connect to the cluster.	Best option for network switches that support unicast MAC.
Multicast MAC	Supported by many switch vendors.	Must configure switches directly attached to the cluster. Might also need to configure all switches on the same subnet as the cluster. For switches that handle multicast as broadcast, you must use filters to restrict traffic to the cluster.	Best option for network switches that do not support unicast MAC.

All nodes in the cluster share a unicast IP address called the c*luster IP address* and a unicast or multicast MAC address called the *cluster MAC address*. The nodes in the cluster receive incoming packets from the *cluster IP,* which is mapped to the multicast or unicast *cluster MAC* address to enable all Sensors to receive a copy of all network traffic.

- Cluster IP addresses. The cluster IP is the gateway that enables network devices to communicate with the HA cluster. When network devices pass traffic to the cluster IP, any node in the cluster can handle the traffic.

NOTE: The cluster IP is not reachable when deployed in Proxy-ARP mode.

- Cluster MAC addresses. You must configure your network switches to forward any packet with the cluster MAC as the destination to all nodes in the cluster.

You must manually configure your network switch to pass multicast or unicast MAC traffic to the IDP Sensors in the HA cluster. For instructions on configuring your switch, see “Switch Hardware and Configuration” on page 163, or consult your switch manufacturer’s operating manual.

Heartbeats

The nodes in an HA cluster communicate with each other using Heartbeat protocol, which uses a multicast IP address. Switches can automatically learn about the Heartbeat protocol using Internet Group Management Protocol (IGMP). You must enable IGMP on your network switch to pass heartbeats to the IDP Sensors in the HA cluster. For instructions on configuring your switch, consult your switch manufacturer’s operating manual.

NOTE: Switches that do not support IGMP cannot be used for a standalone HA configuration and are not supported. See “Standalone HA Switch Compatibility” on page 161 for a list of switches that support IGMP and for additional information about configuring specific network switches.

To decode heartbeat packets, use the Juniper Networks patch for tcpdump that decodes the heartbeat packet data. This patch has been applied to the version of tcpdump that is installed on the Sensors; the patch is also available on the Juniper Networks Support website:

http://www.juniper.net/support/

Use the following guidelines when decoding heartbeat packets with TCPdump:

- Heartbeats go out UDP/5505. Outbound heartbeat packets appear to use the interface of the default route. In reality, however, they use the correct interface (heartbeat interface). This distortion is due to the IDP kernel module.
- The default snaplen (-s) is 68 bytes, which is not enough to decode the entire heartbeat packet. Minimum heartbeat packet size is 112 bytes.
- To verify the MD5 checksum, compile tcpdump with libcrypto support. Pass the command -H < secret > to tcpdump and capture the entire heartbeat packet.

- Heartbeat decoding has three levels:
  - Terse decodes only basic information (default setting).
  - Verbose (-v) decodes the entire heartbeat header.
  - Very Verbose (-vv) decodes the entire heartbeat header as well as the physical and cluster interface information.

Example: Using tcpdump to See Heartbeats

Using the Juniper Networks tcpdump patch, enter the following command:

```
tcpdump -s 0 -tnv -H <secret> -i any udp port 5505
```

This command shows all heartbeats for UDP/5505 on the IDP Sensor that has a Sensor ID of 0.

Choosing Your HA Configuration

Choose a forwarding option for the HA configuration based on your existing network hardware and configuration.

	Router Mode with Layer 3 devices (routers, servers, etc.)...		Proxy-ARP Mode with Layer 3 devices (routers, servers, etc.)...	
If Your Network Switch Supports:	**...can learn multicast ARP**	**...cannot learn multicast ARP**	**...can learn multicast ARP**	**...cannot learn multicast ARP**
Unicast traffic to multiple ports	YES	YES (Best)	YES	YES (Best)
Multicast traffic to multiple ports	YES	Not Recommended (See below)	YES	Not Recommended (See below)

Using Router Mode with Unsupported Multicast ARP

Juniper Networks recommends that you do not use Router mode if your Layer 3 network devices (routers, servers, hosts) do not support multicast ARP. However, if your switch supports only multicast and your devices cannot learn multicast ARP, you must manually add static ARP entries for the cluster IP address to each Layer 3 non-multicast device.

You must enter one ARP entry per non-multicast device. A sample network diagram and the static ARP entry calculations for Proxy-ARP mode is shown on the next page.

Using Proxy-ARP Mode with Unsupported Multicast ARP

Juniper Networks recommends that you do not use Proxy-ARP mode if your Layer 3 network devices (routers, servers, and hosts) do not support multicast ARP. However, if your switch supports only multicast and your devices cannot learn multicast ARP, you must manually add static ARP entries for the IP address of each non-multicast device on each side of the HA cluster.

You must enter an ARP entry for every non-multicast device that exists on each side of the HA cluster; if you have several devices, the number of required ARP entries could potentially be very large. A sample network diagram and the static ARP entry calculations for Proxy-ARP mode is shown here.

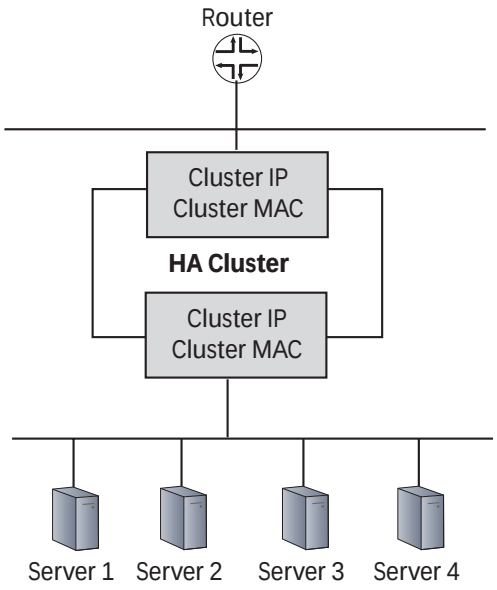
Number of Side A devices that do not support multicast ARP
+
Number of Side B devices that do not support multicast ARP
=
Number of required static ARP entries

Example shown: 1 + 4 = 5

Proxy-ARP Mode Static ARP Entries

Number of Side A devices that do not support multicast ARP times number of Side B devices
+
Number of Side B devices that do not support multicast ARP times number of Side A devices
=
Number of required static ARP entries

Example shown: (1 x 4) + (4 x 1) = 8



Side B has 4 non-multicast devices

Logging for High Availability Events

The IDP Sensor automatically generates log records for HA events; these event appear in the Log Viewer. You can also enable local logging on the IDP Sensor for debugging purposes.

Viewing HA Log Records

The HA daemon also creates log records for important HA events. These log records appear in the Log Viewer and can help you determine the status of your HA cluster.

IDP generates the following HA log records:

- HA_DISABLED. The IDP kernel module has been reconfigured to disable standalone HA. The IDP is handling all incoming traffic.
- HA_WAIT_HA. The IDP kernel module is waiting for schad to start. The IDP does not forward traffic until the HA daemon starts.
- HA_INIT_FORWARD. Schad has started. The IDP is forwarding traffic.
- HA_INIT_BLOCK. Schad has started. The IDP is not forwarding traffic.
- HA_OKAY. Schad has moved from the INIT state to the OK state. The IDP is forwarding traffic.
- HA_LOCAL_FAILED. Schad is experiencing difficulties trying to reach a Verify IP. The link or the Verify IP host may be down. To resolve, look at the destination address.

- HA_REMOTE_FAILED. A remote node in the HA cluster has failed and has either sent a heartbeat FAIL message or is no longer sending heartbeats. To resolve, look at the destination port number for the Node ID of the remote node.
- HA_SHUTDOWN. Schad has shut down with no problems. The IDP is no longer forwarding traffic.

Enabling Local Logging

By default, the HA daemon logs to /var/idp/device/sysinfo/logs/schad.<date>. The verbosity of the logging is based upon the log level, which is set using the SC_DEBUG_LEVEL environment variable.

The HA daemon supports the following levels:

- error. Logs only critical errors.
- warn. Logs errors and other noncritical warning messages (default setting).
- debug. Logs all errors, noncritical warning messages, and internal debugging messages. Use this level only if you need to diagnose a specific problem.

To enable local logging, contact Juniper Networks technical assistance.

Determining HA Status

After you have configured your IDP Sensor in high availability (HA) mode, you should verify that each node is functioning as expected. You can use the command line or the UI to verify HA status, as detailed below.

Viewing HA Status from the Command Line

You can verify HA cluster connectivity using sctop commands at the Sensor command line. Perform the following connectivity test after you have added the second Sensor to the HA cluster and repeat for each subsequent Sensor added:

1. From the Sensor command line, enter the sctop command. The sctop menu appears.
2. Enter w to select HA status. The node and cluster statistics for the IDP Sensor appear.
   - UP indicates that the node is functioning normally.
   - DOWN can indicate that the node is not sending heartbeats, that the node is not receiving heartbeats, or that the switch is experiencing problems.

Viewing HA Status in NSM

You can also verify HA cluster connectivity in the IDP Cluster Monitor component of the UI. In the UI navigation tree, select Realtime Monitor > IDP Cluster Monitor.

System Restart Process

The restart process of IDP Sensors in standalone HA configurations differs slightly from the restart process of a single IDP Sensor. In the restart sequence described below, the HA differences are in bold:

1. The system restarts kernel.
2. The system goes into multi-user mode (run level 3).
3. The lkmStart.sh script inserts the IDP kernel module into the kernel.
4. The lkmStart.sh script tells the IDP kernel that it is using schad. The kernel does not process traffic.
5. The lkmStart.sh script configures the Sensor.
6. The HA daemon (schad) starts.

Schad starts and immediately sends heartbeat packets to the other nodes and attempts to verify the state of all its local interfaces using Verify IP. When schad determines the local state, it enters OK or FAIL mode. If OK, schad tells the IDP kernel module to start processing traffic; if FAIL, the IDP kernel module continues to not process traffic.

Implementing Standalone High Availability

To choose the best deployment mode for your network, determine which type of traffic your existing network devices can pass or be configured to pass. Your network devices need to pass traffic to (or through) the Sensor using a forwarding method (unicast or multicast).

To choose the placement of your IDP Sensors, determine your existing network configuration of IP addresses and subnetworks. You must assign IP addresses to all the IDP Sensor interfaces to allow your network to see the HA cluster.

This section uses several example network diagrams to help you get started with your HA deployment. The diagrams are examples only; you can choose to use different interfaces for forwarding.

NOTE: The IDP Sensor does not support cross-bar configurations that use redundant forwarding mechanisms for a single interface.

Choosing a Deployment Mode

The IDP system supports standalone HA configurations in Proxy-ARP and Router modes. The standalone HA solution can use two different forwarding options to send and receive traffic: Layer 2 unicast or Layer 2 multicast. You choose one of these forwarding options based on your existing network hardware and configuration.

Review your existing network hardware and use the forwarding option table in “Choosing Your HA Configuration” on page 150 to determine the best forwarding method to use for your HA cluster. You are prompted to specify the forwarding method for the standalone HA solution during the Sensor configuration process using the ACM.

To use a network switch that supports only multicast (and not unicast) with network devices that cannot pass multicast ARPs, you must manually configure static ARP entries for each device that cannot pass multicast ARPs. For more information on how to determine the number of static ARP entries you must make, see “Using Proxy-ARP Mode with Unsupported Multicast ARP” on page 150.

NOTE: In Proxy-ARP and Router modes, if you are using multiple subnets in your protected network, you must configure static routes on the IDP Sensor to those subnets. Without these static routes, incoming traffic to those subnets can be lost. Alternatively, you can create a static route from the IDP Sensor to an internal gateway that contains inbound routes to the protected subnets.

Determining Interfaces and Networks

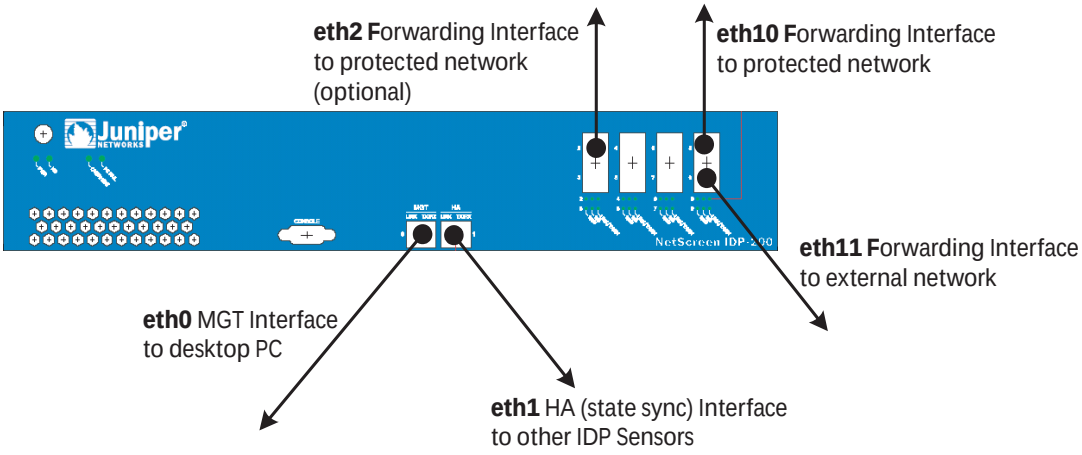
A standalone HA configuration requires three networks: one state-sync network and two forwarding networks, one of which can also be the management network. The IDP Sensor connects to these networks through cables attached to one or more of its interfaces. During the Sensor configuration process, you are prompted to assign IP addresses on these networks to interfaces on the IDP Sensors.

- Forwarding Networks and Interfaces. The interfaces that connect the IDP Sensor to the external network and protected network are forwarding interfaces. You use the forwarding interfaces to send and receive network traffic. You can choose multiple forwarding interfaces on the IDP Sensor; however, the forwarding interfaces on the protected network must use IP addresses that are different from the Management IP address so that the IDP Sensor can route correctly.

- The State-Sync [High Availability (HA)] Network and Interface (Dedicated Network). The interface that connects IDP Sensors to each other is an HA interface. The HA interface is used for communication between the IDP Sensors. State-sync networks can be any dedicated network, including RFC1918 nonroutable networks.

- The Management Network and Interface. The interface that connects to the Management Server is the Management Interface. You use the management interfaces to manage the IDP Sensor from the UI. You must assign the management interface a routable IP address for that network segment.

A sample network/interface configuration is shown here:

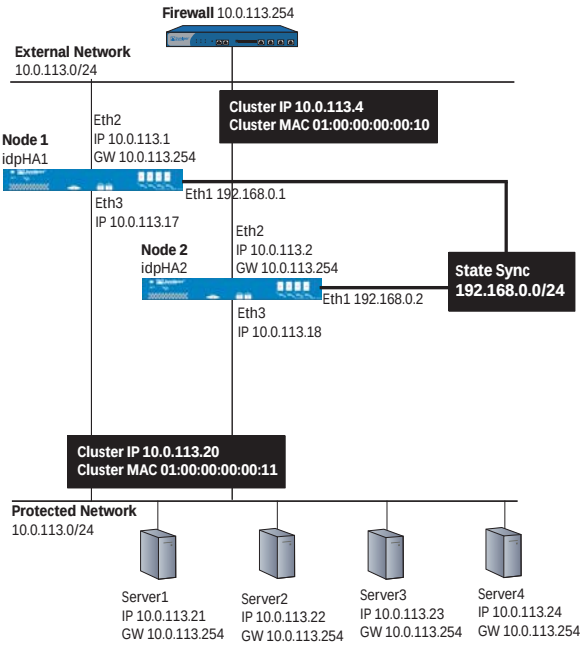


Example Configurations

Use the example configurations in this section to help you determine which IP addresses and networks to use when installing and configuring the IDP system. Each example includes a network diagram with IP and MAC addresses for the HA cluster and the settings for that cluster.

Multicast Proxy-ARP

The following figure shows the network connections for IDP Sensors in Proxy-ARP mode in an HA cluster that uses multicast.



Cluster Settings

The configuration shown in the previous diagram uses the following cluster settings. You configure these cluster settings during the Sensor configuration process using ACM. All Sensors in the HA cluster use the same HA cluster settings to enable schad communication. These cluster settings determine how the HA daemon verifies and manages the state of the cluster.

Schad Communication	
Shared secret	N3t5cr33n
Cluster ID	0
Total nodes	3
Interval	1 second
Fail count	3
Reintegrate count	5

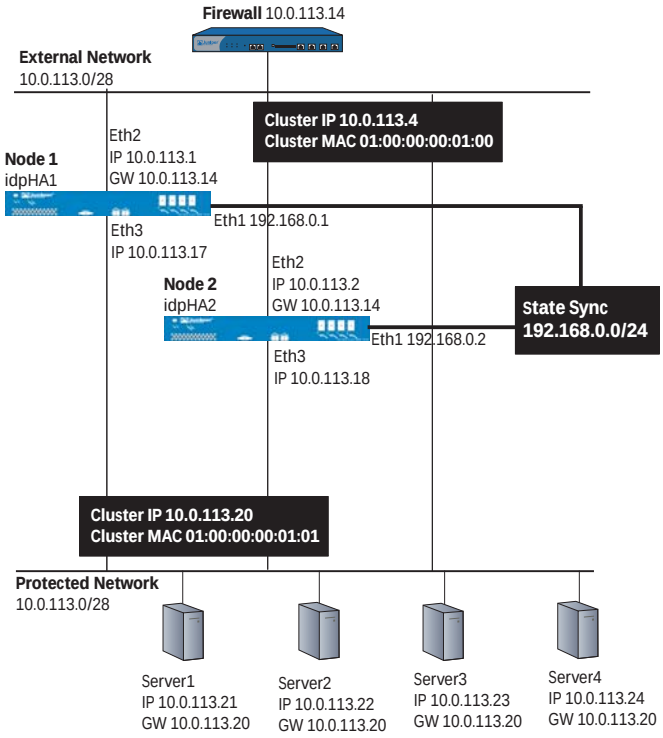
Interface Settings

The configuration shown in the previous diagram uses the following interface settings. All Sensors in the HA cluster use identical interface settings for a given interface. For example, all eth2 interfaces on all Sensors in the HA cluster use the same interface settings.

eth2		eth3	
cluster ip	10.0.113.4	cluster ip	10.0.113.20
cluster mac	01:00:00:00:00:10	cluster mac	01:00:00:00:00:11
verify ip	10.0.113.254	verify ip	10.0.113.21
heartbeat ip	239.0.0.10	heartbeat ip	239.0.0.11

Multicast Router

The following diagram shows the network connections for IDP Sensors in Router mode in an HA cluster that uses multicast.



Cluster Settings

The example configuration uses the following cluster settings:

Schad Communication	
shared secret	N3t5cr33n
cluster ID	0
total nodes	3
interval	1 second
fail count	3
reintegrate count	5

Cluster settings determine how the HA daemon verifies and manages the state of the cluster. To enable schad communication when you configure the IDP Sensor using ACM, you must configure all Sensors in the HA cluster use the same HA cluster settings.

Interface Settings

The configuration shown in the previous diagram uses the following interface settings. All Sensors in the HA cluster use identical interface settings for a given interface. For example, all eth2 interfaces on all Sensors in the HA cluster use the same interface settings.

eth2		**eth3**	
cluster ip	10.0.113.4	cluster ip	10.0.113.20
cluster mac	01:00:00:00:01:00	cluster mac	01:00:00:00:01:01
verify ip	10.0.113.14	verify ip	10.0.113.21
heartbeat ip	239.0.0.10	heartbeat ip	239.0.0.11

Multicast Proxy-ARP with Juniper Networks Firewalls

For maximum failover protection, you can deploy an HA cluster of IDP Sensors in Proxy-ARP mode behind Juniper Networks firewalls that use an active/active configuration.

Your Juniper Networks firewalls must support NSRP (NetScreen-50 firewalls and higher models) and run ScreenOS 4.0.0 or higher. NetScreen-5XT, NetScreen-5XP, and NetScreen-25 firewalls do not support NSRP and cannot be used for IDP HA. Juniper Networks recommends that you use NetScreen-208 or higher models for an active/active configuration with an IDP HA cluster.

You must also use a network switch that supports IGMP to connect your Juniper Networks firewalls and IDP Sensors.

To deploy an IDP HA cluster in Proxy-ARP with Juniper Networks firewalls in active/active:

1. Setup and configure your Juniper Networks firewalls using the documentation that came with your firewall. Set NSRP to active/active, and configure two virtual security devices (VSD) for each Juniper Networks firewall:

   - Configure each Juniper Networks firewall as master of one VSD and primary backup of the other VSD.

   - For each physical interface (including trust zone and untrust zone), assign a unique IP address to each VSD. This interface is now a virtual security interface (VSI).

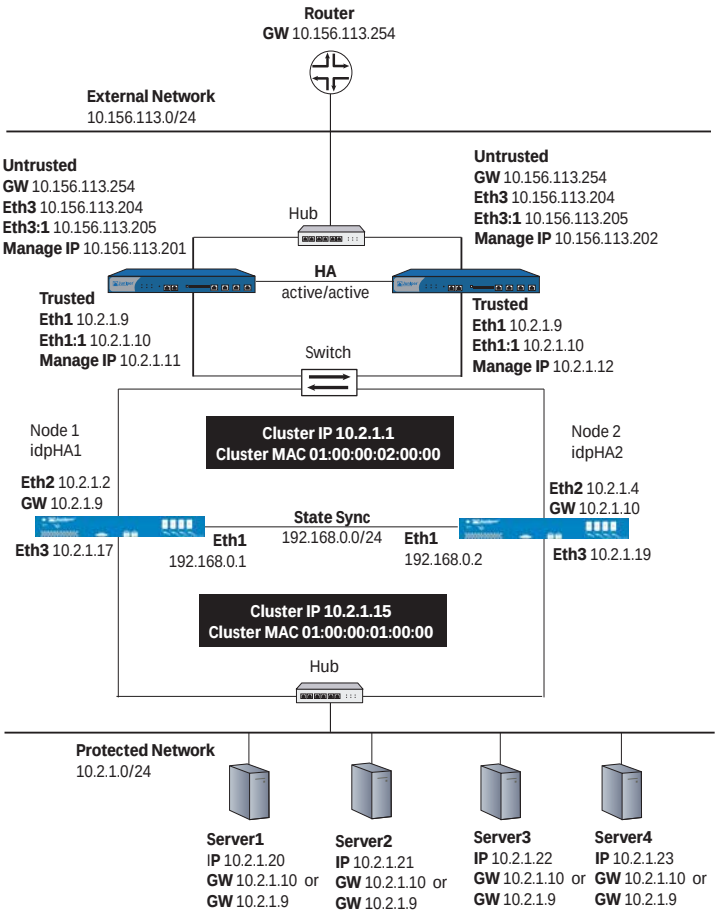
   - Each Juniper Networks firewall must use the same settings for each VSI. (The manage IP addresses, which are used for management purposes only, can be different).

2. Connect the IDP Sensors to your network. Using the ACM, configure the Sensor software on each IDP Sensor to use Proxy-ARP mode, standalone HA with load balancing, and multicast.

   - The gateway for node 1 is the trusted VSI of the first VSD.
   - The gateway for node 2 is the trusted VSI of the second VSD.
   - The gateway for hosts on the protected network can be the trusted VSI of first or second VSD.

NOTE: In Proxy-ARP and Router modes, if you are using multiple subnets in your protected network, you must configure static routes on the IDP Sensor to those subnets. Without these static routes, incoming traffic to those subnets can be lost. Alternatively, you can create a static route from the IDP Sensor to an internal gateway that contains inbound routes to the protected subnets.

The following figure shows the network connections for this configuration:



Cluster Settings

The example configuration uses the following cluster settings:

Schad Communication	
shared secret	N3t5cr33n
cluster ID	0
total nodes	2
interval	1 second
fail count	3
reintegrate count	5

Cluster settings determine how the HA daemon verifies and manages the state of the cluster. To enable schad communication when you configure the IDP Sensor using ACM, you must configure all Sensors in the HA cluster use the same HA cluster settings.

Interface Settings

The example configuration uses the following interface settings:

eth3		eth2	
cluster ip	10.2.1.1	cluster ip	10.2.1.15
cluster mac	01:00:00:02:00:00	cluster mac	01:00:00:01:00:00
verify ip (node 1)	10.2.1.9	verify ip (node 1)	10.2.1.20
verify ip (node 2)	10.2.1.10	verify ip (node 2)	10.2.1.20
heartbeat ip	239.0.0.10	heartbeat ip	239.0.0.11

All Sensors in the HA cluster use identical interface settings for a given interface, *except* for the eth3 Verify IP:

- For node 1, the eth3 Verify IP is 10.2.1.9, which is the trusted VSI of the first VSD.
- For node 2, the eth3 Verify IP is 10.2.1.10, which is the trusted VSI of the second VSD.

Note that the eth3 Verify IP for each IDP Sensor is also the default gateway for that IDP Sensor.

Installing Your IDP Sensors

After you have chosen a deployment mode, determined which interfaces you want to use on your IDP Sensors, and reviewed the provided examples, you are ready to install your IDP system.

Reconfiguring Standalone HA Clusters

To reconfigure your standalone HA configuration using the ACM:

1. For all cluster nodes except the first node, stop the IDP processes. From the Sensor command line, enter the following command:

   service idp stop

2. You must also stop sctop processes, if they are running. From the Sensor command line, enter the following command:

   quit sctop

3. Use the ACM on the first cluster node to change the standalone HA configuration as desired, then save and apply the configuration.

4. Use the ACM on all other cluster nodes to change the standalone HA configuration to match the configuration on the first node, then save and apply the configuration.

All cluster nodes should now be running the IDP processes. To verify the status of the cluster, see "Determining HA Status" on page 152.

Standalone HA Switch Compatibility

Table 20 lists switches that are compatible with the IDP standalone high availability (HA) solution.

NOTE: Some switches might work even though they are not recommended.

Table 20: Compatible Switches

Vendor	Product	SW/FM Version	Cluster Forwarding Mode			Notes	Tested	Rcmd
			IGMP Piggyback[1]	Multicast MAC	Unicast MAC			
Foundry	FastIron Workgroup	7.5	Yes	Yes[2]	No	Requires multicast filters.	No	Yes
Foundry	FastIron Workgroup	6.x	Yes	No[2]	No	Upgrade to 7.5 SW.	No	No
Foundry	BigIron	7.2	Yes	No[2]	No	7.5 SW should work.	No	[b]No
Foundry	ServerIron	7.2.10T22	Yes	Yes	Yes		Yes	Yes
Nortel/Alteon	AceDirector AD3		Yes	No	No		No	No
Nortel/Alteon	AceDirector 184e		Yes	Yes	No		No	Yes
Cisco	Catalyst 3500XL	12.0(5)WC7 (released 03-05-2003)	Yes[2]	No	No		No	No
Cisco	Catalyst 3550	IOS 12.1	Yes	No	Yes		Yes	No
Cisco	Catalyst 4006	IOS 12.1	Yes	Yes	Yes		Yes	Yes
Cisco	Catalyst 4000	CatOS 6.1	Yes	Yes	Yes		Yes	Yes

Vendor	Product	SW / FM Version	Cluster Forwarding Mode			Notes	Tested	Rcmd
			IGMP Piggyback[1]	Multicast MAC	Unicast MAC			
Dell	PowerConnect 3524		Yes[2]	No	No	Tested.	Yes	No
Extreme	Extreme Summit 24e3	Extremeware 6.2e.1 (build 15), released 08-13-2002	No	No[2]	No[2]	Juniper Networks has tested and confirmed that this switch does not work.	Yes	No
Cisco	Catalyst 5xxx/6xxx Series		Yes[2]	Yes[2]	No[2]	Stonesoft docs indicate these switches work.	No	Yes
Cisco	Catalyst 2900XL	IOS 12.0 (5.1) XP, released 12-10-1999	Yes[2]	Yes[2]	Yes[2]	Juniper Networks has tested the Cisco 2950 and confirmed that this switch does not work. Other 2900 series should work.	Yes	Yes
Nortel/Bay	BayStack 450		Yes[2]	No[2]	No[2]	Stonesoft docs indicate this switch works.	No	No
Nortel	Business Policy Switch 2000		No[2]	No[2]	No[2]	Stonesoft docs indicate this switch works.	No	No
Nortel	Passport 1200		No[2]	No[2]	No[2]	Stonesoft docs indicate this switch works.	No	No
Nortel	Passport 8100		No[2]	No[2]	No[2]	Stonesoft docs indicate this switch works.	No	No
Nortel	Passport 8600		No[2]	No[2]	No[2]	Stonesoft docs indicate this switch works.	No	No

1.This mode requires setting the cluster MAC for a given interface to the same MAC used by the IP multicast address (also used for heartbeats). This mode is not recommended because of potential ARP compatibility issues with other devices on the network. Unless otherwise noted, the switch has not been tested in this mode.

2.See the Notes column for details.

Switch Hardware and Configuration

Standalone HA configurations require you to configure your network switch to pass Layer 2 multicast or unicast MAC packets and heartbeats to the HA cluster. You choose a forwarding option (unicast or multicast) based on your existing network hardware and configuration.

Layer 2 Unicast/Multicast

The standalone HA solution uses two types of multicast traffic:

- IP multicast is used to pass heartbeat packets between nodes.
- MAC (Layer 2) multicast is used to forward network traffic to the networks on each side of the HA cluster. Devices on the network that want to talk to a device on the other side of the HA cluster use the cluster MAC address in the destination field of the Ethernet frame. Additionally, the nodes in an HA cluster receive incoming packets via a *cluster IP*, which is mapped to a multicast or unicast *cluster MAC* address to enable all Sensors to receive a copy of all network traffic.

To enable nodes in an HA cluster to communicate with network on each side of the cluster and with other nodes, you must perform the following steps:

- Manually configure your network switches to pass Layer 2 multicast or unicast MAC traffic to the IDP Sensors in the HA cluster.
- Enable IGMP on your network switches to pass heartbeats to the IDP Sensors in the HA cluster.

Forwarding Switch Requirements

Switches that connect to the IDP Sensors' forwarding interfaces must be able to forward traffic to the cluster MAC address. The switch must be able to learn or be configured to pass multicast MAC traffic to the Sensors in a "limited broadcast." Enterprise-class switches can accomplish this in three ways:

- The administrator hard codes the unicast cluster MAC to all switch ports that attach to an IDP Sensor.
- The switch broadcasts multicast cluster MAC traffic to all ports in the VLAN. Administrators can apply filters to limit the broadcast on some switches; however, some enterprise-class switches do not allow filtering. *Juniper Networks highly recommends that you do not use a multicast broadcast-only switch in your HA configuration, as this can affect network performance and reduce overall network security.*
- The switch passively learns which ports should forward multicast traffic to the IDP Sensor based upon the source and destination MAC address of traffic flowing through the port. However, this passive mechanism is not standardized across switch vendors.

NOTE: This mechanism differs from IGMP, which is a standardized, active notification method.

- The forwarding switch must also pass multicast IP traffic (heartbeat packets) to the other nodes in the HA cluster. To accomplish this, the administrator can enable IGMP on the network switch. IGMP is a protocol that enables switches to automatically learn about devices that belong to an IP multicast group. Some switches have IGMP enabled by default.

State-Sync Switch Requirements

You can use a cross-over cable or switch to connect the IDP Sensor state-sync interfaces, depending on the number of Sensors you are using in the HA cluster.

- If you are using only two IDP Sensors in an HA cluster, you do not need to used a switch to connect them. Instead, use a cross-over cable to provide connectivity for the state-sync mechanism. This method is more cost-effective and robust.
- If you are using more than two IDP Sensors in your HA cluster, you must use a network switch to provide connectivity for the state-sync mechanism.

The state-sync switch does not require special features. However, for gigabit networks or more than three IDP Sensors in an HA cluster, you should use a switch that has a dedicated store-and-forward cache for each port for best performance.

Because this state-sync switch represents a single point of failure for node communication, *Juniper Networks highly recommends that you use a reliable switch manufactured by a reputable vendor.* As a precaution, you might want to keep a cold spare available should the state-sync switch fail.

VLANs

Because the security features of VLAN implementations differ, Juniper Networks recommends that you do not use VLANs to separate traffic on each node of the HA cluster. Use a dedicated switch instead. For more information on VLANs, see “Using Tagged Interfaces and Virtual Routers” on page 135.

Switch Hardware

This list is not comprehensive of every switch or vendor that does or does not work in a standalone HA configuration. Different versions or branches of the switch software might behave differently than those devices tested.

Tested Switches

Juniper Networks tested several switches in standalone HA configurations. IDP supports the following switches with the proper configuration:

- Foundry FastIron Stackable series (multicast with filtering). Juniper Networks tested Foundry FastIron 400, IronWare 07.6.03 (released 05-08-2003).
- Foundry BigIron Chassis series (multicast with filtering)

- Cisco Catalyst (IOS series) switches (unicast or multicast with filtering). Juniper Networks tested:
  - Cisco Catalyst 3500XL, 12.0(5)WC7 (released 03-05-2003)
  - Cisco Catalyst 2900XL, IOS 12.0 (5.1) XP (released 12-10-1999)
- Cisco Catalyst (CatOS series) switches (multicast with filtering). Juniper Networks tested Cisco Catalyst 6509, CatOS 5.5(1).

The following switches do not work:

- Cisco 2950
- Alteon/Nortel AceDirector series
- 3Com SuperStack II 3300 XM
- 3Com SuperStack III 4400
- Extreme switches. Juniper Networks tested Extreme Summit 24e3, Extremeware 6.2e.1 (build 15) (released 08-13-2002)

Untested Switches

Juniper Networks did not test all switches in standalone HA configurations and might not support older types of switches.

The IDP system supports the following switches with the proper configuration:

- Juniper
- Nortel Passport 1200
- Intel

The following switches might not work:

- Nortel BayStack 450, Nortel Business Policy 2000, Nortel Passport 8100/8600
- Bay Networks
- Cisco CSS series (Arrowpoint)
- RadWare
- Nortel BayStack
- HP Procurve Series. Juniper Networks engineering experience indicates that these switches do not work; however, the manufacturer's operating manual indicates that they do.

If you use an untested switch in your network, Juniper Networks recommends that you contact the switch manufacturer before you implement an IDP HA configuration. The switch manufacturer might be able to provide internal test results, configuration guides, or reference other customer installations to help you implement IDP HA successfully.

Unknown Switches

Juniper Networks did not test the following switches:

- Centillions (now Nortel)
- Shasta (now Nortel)
- Accellars (now Nortel)

Configuring Switches

This section includes details for configuring Foundry, Cisco IOS, Cisco COS, and HP Procurve switches.

Foundry

Juniper Networks tested the following Foundry switches:

- Foundry FastIron 400
- IronWare 07.6.03, released 05-08-2003
- Multicast only
- Proxy-ARP and Router
- Load-Balancing and Hot-Standby

To configure this Foundry switch:

1. Enable passive IP multicast support by typing the command:

   ip multicast passive

   This enables the switch to learn about heartbeat IP Multicast via IGMP.

2. Prevent traffic to the HA cluster from being broadcast to all ports on the VLAN by typing the commands:

   multicast filter 1 any mac *cluster-mac*
   exclude-ports ethernet *list all non-IDP ports here*

   This prevents traffic destined for the cluster from being broadcast to all ports which would otherwise reduce security and performance.

3. Add static ARP entries to the switch ARP table by entering the command:

   arp *num ip address mac-address port*

Because Foundry switches do not automatically learn multicast MAC addresses, you must add the static ARP entries before you can contact the switch through the HA cluster for management, monitoring, or SNMP purposes.

Cisco IOS Series (2900XL / 3500XL)

Juniper Networks tested the following Cisco switches running IOS:

- Cisco Catalyst 3500XL
  - 12.0(5)WC7, released 03-05-2003
  - Multicast and Unicast
  - Proxy-ARP and Router
  - Load-Balancing and Hot-Standby
- Cisco Catalyst 2900XL
  - IOS 12.0 (5.1) XP, released 12-10-1999
  - Multicast and Unicast
  - Proxy-ARP and Router
  - Load-Balancing and Hot-Standby

To configure these switches:

1. Enable IGMP support. In the VLAN interface configuration mode, enter the following command:

   ip igmp snooping

2. Configure a static MAC address by typing the command:

   ip igmp snooping static *cluster-mac interface*

NOTE: The MAC address must use the format 0000.0000.0000.

Some switches might require the alternate format as shown below:

mac-address-table static *cluster-mac source-port destination-ports* **vlan** *vlan-id*

destination-ports are the ports on the IDP Sensor and *source-port* is the port that is sending traffic to the IDP Sensor. For example, a 24-port switch with 22 devices and 2 IDP Sensors requires 22 configuration statements (one per client).

Cisco CatOS Series (2948G / 4000 / 5000 / 5500 / 6500)

Juniper Networks tested the following Cisco switch running CatOS:

- Cisco Catalyst 6509
- CatOS 5.5(1)
- Multicast only (no unicast)
- Proxy-ARP and Router
- Load-balancing and hot-standby

To configure this switch, perform the following steps:

1. Disable CGMP and GMRP.
2. Enable IGMP by entering the command:

   set igmp enable

3. Hard code the cluster MAC by entering the following command:

   set cam permanent *cluster-mac module* **/** *port(s) vlan-id*

NOTE: The MAC address must use the format 00-00-00-00-00-00.

This command tells the switch to send traffic that is destined for the cluster MAC address to the ports listed. IGMP, CGMP, or GMRP must be enabled before you can hard code multicast MAC addresses.

HP Procurve Series

Juniper Networks recommends that you do not use an HP Procurve switch as they cannot keep static MAC address entries. You can configure the HP Procurve to forward all multicast traffic to certain ports, but performance and security might suffer. By default, IGMP is disabled on HP Procurve switches.

1. Enable IGMP and multicast traffic forwarding by typing the command:

   vlan *vlan-id* **ip igmp forward** *forward-ports* **auto** *heartbeat-ports*

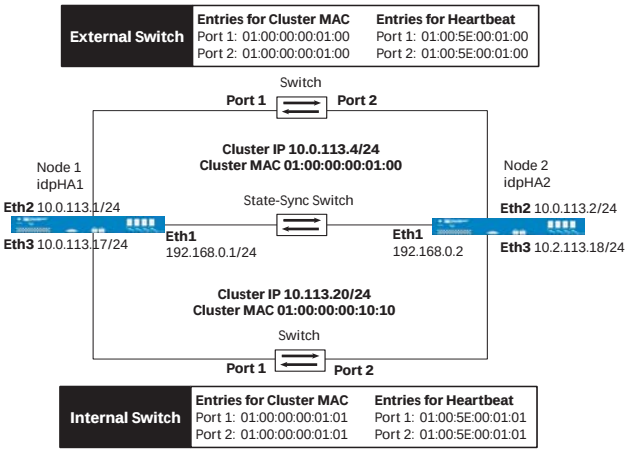
   forward-ports is a comma separated list of ports that connect to the forwarding or heartbeat interfaces on the IDP Sensor. *heartbeat-ports* are heartbeat-only ports.

2. Assign a high priority to multicast traffic (to improve performance) by typing the command:

   vlan *vlan-id* **ip igmp high-priority-forward**

Sample Switch Configuration

A sample switch configuration is shown in the following figure.



External High Availability

You can also deploy IDP Sensors in high availability (HA) configuration using external HA devices. In HA mode, two or more IDP Sensors are paired to provide failure protection and/or load balancing. The IDP external HA solution requires:

- Two or more IDP Sensors, running in Bridge, Transparent, or Router mode, that are connected to each other. (To connect two IDP Sensors, use a crossover cable. To connect more than two IDP Sensors, use a hub or switch.)
- An external high availability hardware solution that provides failure protection and/or load balancing. When you purchased IDP, your sales engineer gave you information about the load balancers and firewall Sensors that are compatible with the IDP external HA feature.

This chapter uses several example network diagrams to help you get started with your HA deployment. The diagrams are examples only; you can choose to use different interfaces for forwarding, management, and state-sync depending on your IDP Sensor and your existing network configuration.

Deployment Modes

You can deploy the IDP Sensors in high availability using two methods:

- As an active-gateway bridge using two Juniper Networks firewall Sensors that support NSRP.
- As an active-gateway router using two load balancers

To choose the deployment mode for your network, determine your existing external HA failure protection and/or load balancing hardware, then determine how your existing network configuration of IP addresses and subnetworks is currently constructed.

Review the information in the following sections and choose a deployment mode that works with your existing hardware.

Determining Interfaces and IP Addresses

A external HA configuration requires three networks: two forwarding networks and one state-sync network. The management network can use the forwarding interface that faces the protected network.

The IDP Sensor connects to these networks through cables attached to one or more of its interfaces. During the Sensor configuration process, you are prompted to assign IP addresses on these networks to interfaces on the IDP Sensors:

- Forwarding Networks and Interfaces. The interfaces that connect the IDP Sensor to the external network and protected network are forwarding interfaces. Forwarding interfaces send and receive network traffic.

  You can choose multiple interfaces on the IDP Sensor as forwarding interfaces. However, to increase performance, Juniper Networks recommends that you assign forwarding interfaces to those interfaces that share a network driver: Use eth2 and eth3 as forwarding interfaces, or use eth4 and eth5 as forwarding interfaces.

- The State-Sync Network and Interface (Dedicated Network). The state-sync interface connects the IDP Sensors. The HA daemon uses the dedicated state-sync interface to synchronize traffic flow between the IDP Sensors. State-sync IP addresses can be any IP address, including RFC1918 nonroutable IPs.

- The Management Network and Interface. The interface that connects to the Management Server is the Management Interface. Use the Management Interface to manage the IDP Sensor from the UI. Choose one interface as the Management Interface and assign it a routable IP address for that network segment.

  To use a forwarding interface as the Management Interface, you can assign the interface an IP address. However, because of security risks, Juniper Networks does not recommend this option. Instead, use a dedicated interface for management—this option decreases the risk that an attacker might be able to communicate directly with the IDP Sensor.

Deploying with Juniper Networks Firewalls

You can use two Juniper Networks firewalls to provide hot standby external HA for IDP Sensors running in Bridge or Transparent mode. Your Juniper Networks firewalls must support NSRP (NetScreen-50 firewalls and higher models) and be running ScreenOS 4.0.0 or higher. NetScreen-5XT, NetScreen-5XP, and NetScreen-25 firewalls do not support NSRP and cannot be used for IDP HA.

The Juniper Networks firewalls can detect IDP failure using the following mechanisms:

- To detect failure in the connection between the Juniper Networks firewall and the IDP Sensor, each Juniper Networks firewall monitors the status of its own interfaces. If the untrusted or trusted interface goes down, the Juniper Networks firewall initiates hot standby failover to the other Juniper Networks firewall, forcing all sessions to pass through the other IDP Sensor.

- You can also configure each Juniper Networks firewall to use Track IP to monitor any connection failure between the Juniper Networks firewall and the Track IP hosts, which can be on the opposite side of the IDP Sensors (For example, the IDP Management Server).

  If the value of the Track IP failure exceeds the user-specified threshold, the Juniper Networks firewall considers itself to be down and initiates failover to the other Juniper Networks firewall. All sessions (old sessions initiated before the failover and new sessions initiated after the failover) are forced to pass through the other IDP Sensor.

To deploy with Juniper Networks firewalls, perform the following steps:

1. Set up and configure your Juniper Networks firewalls using the documentation that came with your firewall.

2. Set NSRP to active/passive.

3. Connect the IDP Sensors to your network.

4. Using the ACM, configure the Sensor software on each IDP Sensor to use Bridge or Transparent mode, external HA, hot standby:

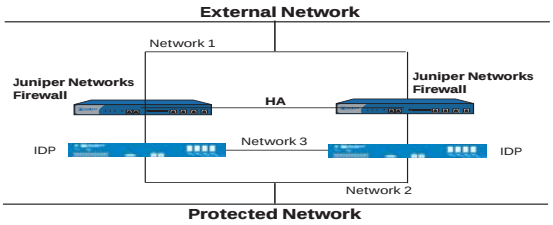
   - You must use a stealth interface (no assigned IP address) for the forwarding interface that does not connect to the IDP Management Server.

   - You must assign an IP address to the Management Interface on each IDP Sensor. If you are using a forwarding interface as a management interface, you must assign that interface an IP address.

   - If you are using a dedicated Management Interface, you can use a stealth interface for all forwarding interfaces on the IDP Sensor.

Network Configuration

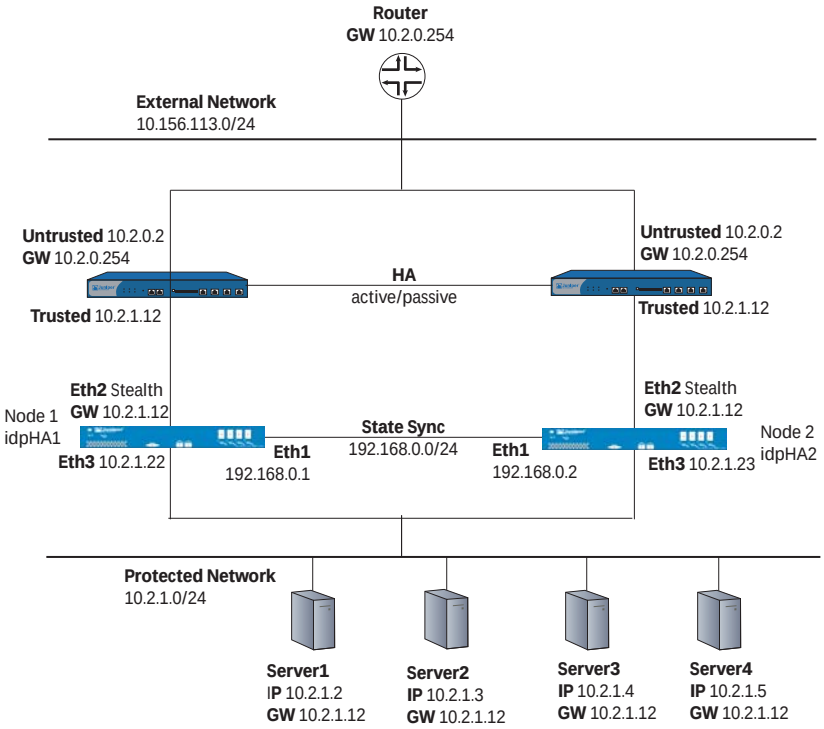
The following figure shows an example network configuration.



- Network 1 connects the external network to the Juniper Networks firewalls.
- Network 2 connects the protected network to IDP and Juniper Networks firewalls.
- Network 3 connects the IDP Sensors to one another.

IP Configuration

The figure below shows an example IP configuration.



The previous figure displays a dedicated Management Interface with an IP address for each IDP Sensor. Both forwarding interfaces for each IDP Sensor are stealth interfaces, indicating that they do not have an assigned IP address.

Deploying with Load Balancers

You can use two load balancers to provide load balancing external HA for IDP Sensors running in Router mode. For instructions on setting up and configuring your load balancers, consult the vendor documentation that came with your load balancer.

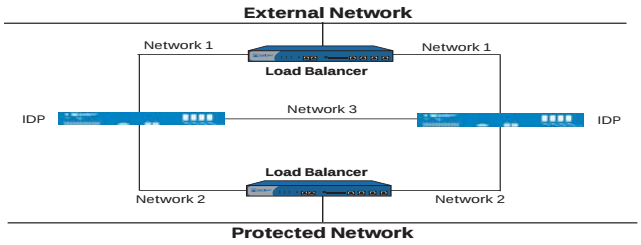
NOTE: Any Open Platform for Security (OPSEC)-compatible, hardware-only load balancing solution should work.

To deploy with load balancers:

1. Set up and configure your load balancers using the vendor documentation.
2. Connect the IDP Sensors to your network.
3. Using the ACM, configure the Sensor software on each IDP Sensor to use Router mode, external HA, load balancing. You must also assign IP addresses for each network that connects to the IDP.

Network Configuration

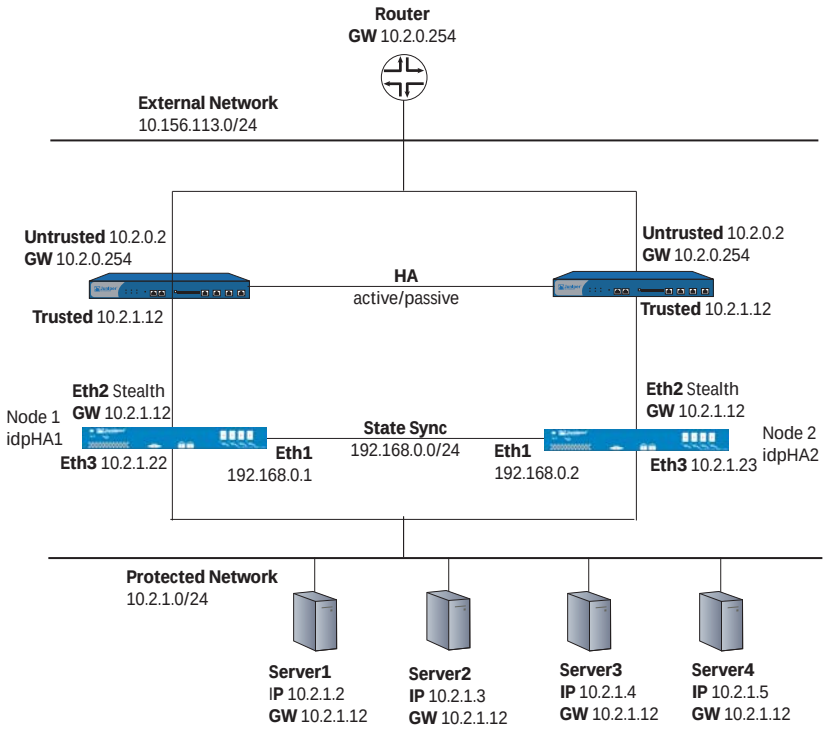
The following figure shows an example network configuration.



- Network 1 connects the external network and load balancer to the IDP Sensors.
- Network 2 connects the protected network and load balancer to the IDP Sensors.
- Network 3 connects the IDP Sensors to one another.

IP Address Configuration

The following figure shows an example IP configuration.



Spanning Tree Protocol

IDP supports Spanning Tree Protocol (STP) for IDP Sensors running in Bridge or Transparent mode:

- IDP Sensors in Bridge mode can actively participate in STP.
- IDP Sensors in Transparent mode do not actively participate in STP, but they do pass the BPDUs used by STP.

Enabling STP

To enable STP, edit the user_funcs file on the Sensor:

1. From the Sensor command line, open /usr/idp/device/bin/user_funcs in a text editor.
2. Locate the line user_start_end() and add the following line below it:

   $SCIO const -v vr0 set sc_stp_enabled 1

3. Save the file and exit the editor.
4. Reboot the Sensor.

Monitoring STP

To monitor the STP state for each forwarding interface, use the sctop command line utility with option p.

Disabling STP

To temporarily disable STP until the next Sensor reboot or until you restart the Sensor processes, enter the command:

```
scio const -v vr0 set sc_stp_enabled 0
```

To permanently disable STP, perform the following steps:

1. From the Sensor command line, open /usr/idp/device/bin/user_funcs in a text editor.
2. Locate the line user_start_end() and remove the following line:

   ```
   $SCIO const -v vr0 set sc_stp_enabled 0
   ```

3. Save the file and exit the editor.
4. Reboot the Sensor.

Appendix A

Command Line Utilities

Use command line utilities to manage and troubleshoot the IDP system from the command line interface. Juniper Networks provides the command line utilities shown below:

- idp.sh. Starts, stops, and monitors the status of Sensor processes.
- scio. Configures the IDP system.
- sctop. Monitors connection tables and displays Sensor status.
- bypassStatus. Displays the NIC Bypass (Internal Bypass) state for copper port pairs.
- statview. Displays statistics collected by Application Volume Tracking.

The following sections detail each command line utility location, option, functions, and arguments.

Shell Setup

To use command line utilities, including the log2action command, on the Sensor and IDP Management Server, you should first make sure that the following settings are defined in the shell. bash is the default shell; changing to another shell requires that you use equivalent commands.

```
IDPDIR=/usr/idp
export IDPDIR
```

idp.sh Commands

Use the idp.sh utility to control Sensor processes.

Location	/usr/idp/device/bin	
Syntax	idp.sh options	
Option	Function	Arguments
start	Starts the Sensor.	None
status	Displays the status of the Sensor.	None
stop	Stops the Sensor processes.	None
restart	Restarts the Sensor processes.	None
version	Displays the version of the Sensor processes and kernel.	None
reload	Restarts the Sensor processes.	None

scio Commands

Use scio commands to configure the IDP system. Many scio commands are replicated by the parameters in the Sensor Settings rulebase.

Location	/usr/bin
Syntax	scio [command] [options] [arguments]

Commands for the scio utility are shown below.

agentconfig

Use the agentconfig command to configure NSM communications settings.

Syntax	scio agentconfig options arguments	
Options	Function	Arguments
server1-ip	Displays or sets the IP address of the primary NSM server.	list < IP address >
server1-port	Displays or sets the port of the primary NSM server.	list < port number >
server2-ip	Displays or sets the IP address of the secondary NSM server.	list < IP address >
server2-port	Displays or sets the port of the secondary NSM server.	list < port number >
otp	Displays or sets the one-time password of the Sensor.	list < one-time password >
deviceid	Displays or sets the device ID of the Sensor. The ID must be 42 hexadecimal characters.	list < device ID >

const

Use the const command to set and view kernel constants. Many of these values can also be set in the NSM UI, under Sensor Settings.

Syntax	scio const -s s0 options arguments	
Options	Function	Arguments
get	Displays the value of a constant.	< constant >
set	Specifies the value of a constant.	< constant > < value >
list	Displays all constant–value pairs.	None

GRE Decapsulation Constants

These constants control GRE decapsulation on an IDP Sensor. IDP supports both IP-in-GRE and PPP-in-GRE decapsulation.

sc_gre_decapsulation
Description: Controls whether the Sensor decapsulates GTP traffic.
Possible values: 0-1 (0 = off, 1 = on)

Default value: 0
Example: scio const -s s0 set sc_gre_decapsulation 1

sc_max_decapsulation
Description: Controls how many levels of encapsulation the Sensor will unpack. Applies to all types of encapsulation.
Possible values: 1-2
Default value: 1
Example: scio const -s s0 set sc_max_decapsulation 2

GTP Decapsulation Constants

These constants control GPRS Tunneling Protocol (GTP) decapsulation on an IDP Sensor. IDP supports UDP GTPv0 and GTPv1 only.

sc_gtp_decapsulation
Description: Controls whether the Sensor decapsulates GTP traffic.
Possible values: 0-1 (0 = off, 1 = on)
Default value: 0
Example: scio const -s s0 set sc_gtp_decapsulation 1

sc_gtp_timeout
Description: Controls how long a Sensor holds a GTP tunnel, in seconds, from the time of the last packet. If the time elapses without another packet, the Sensor considers the tunnel closed.
Possible values: 1-0xFFFFFFFF (seconds)
Default value: 3600 (seconds)
Example: scio const -s s0 set sc_gtp_timeout OxFFFA

sc_max_decapsulation
Description: Controls how many levels of encapsulation the Sensor will unpack. Applies to all types of encapsulation.
Possible values: 1-2
Default value: 1
Example: scio const -s s0 set sc_max_decapsulation 2

sc_gtp_max_flows
Description: Controls how many GTP tunnels the Sensor will decapsulate at one time. If
Possible values: 2-0x61A80 (2-400,000)
Default value: 0x30D40 (200,000)
Example: scio const -s s0 set sc_gtp_max_flows 0x30F55

SSL Constants

sc_ssl_decryption
Description: Turns SSL Inspection on (1) or off (0).
Possible values: 0-1
Default value: 0
Example: scio const -s s0 set sc_ssl_decryption 1

sc_ssl_sessid_timeout
Description: SSL session security parameter cache timeout value
Possible values: 1-120 (seconds)
Default value: 60
Example: scio const -s s0 set sc_ssl_sessid_timeout 90

sc_ssl_pending_sessid_timeout
Description: SSL pending session security parameter cache timeout value
Possible values: 1-60 (seconds)
Default value: 30
Example: scio const -s s0 set sc_ssl_pending_sessid_timeout 45

sc_ssl_num_decrypt_sessions
Description: Maximum number of SSL sessions for decryption

Possible values: 1-100000
Default value: 10000
Example: scio const -s s0 set sc_ssl_num_decrypt_sessions 50000

Application Volume Tracking Constants

This constant controls whether Application volume Tracking is turned on or not. See *statview Command* on page 191 for more information on Application Volume Tracking.

sc_periodic_stat_update
Description: Turns AVT on (1) or off (0).
Possible values: 0-1
Default value: 0
Example: scio const -s s0:flow set sc_periodic_stat_update 1

SYN-Protector Constants

These constants control SYN Cookie settings for the SYN-Protector rulebase. You must specify the syndef module when running the commands. See the examples for full syntax.

sc_syndef_timeout
Description: Passive mode only. Controls how many seconds the Sensor will hold an incomplete SYN-ACK handshake before purging it.
Possible values: 1-0xFFFF (seconds)
Default value: 5
Example: scio const -s s0:syndef set sc_syndef_timeout 10

sc_syndef_threshold
Description: Controls the lower threshold for SYN Protection activation. For SYN Cookie (relay) mode, this is the only value that matters. Passive mode also uses the “delta” value below.
Possible values: 1-0xFFFF (SYN packets per second per destination IP)
Default value: 0x3E8 (1000)
Example: scio const -s s0:syndef set sc_syndef_threshold 2000

sc_syndef_threshold_delta
Description: Upper threshold for SYN-Protector activation. As of IDP 4.0, has no meaning for SYN Cookie (relay) mode. For Passive mode, SYN Protection activates once the number of SYN packets per second for a given destination IP exceeds this number plus the lower threshold number. Passive mode protection deactivates once the value drops below the lower threshold.
Possible values: 1-0xFFFF (SYN packets per second per destination IP)
Default value: 0x14 (20)
Example: scio const -s s0:syndef set sc_syndef_threshold_delta 10

sc_syndef_report_freq
Description: Controls how often a SYN flooding attempt is reported, in seconds.
Possible values: 1-86,400 (in seconds) (86,400 seconds is 1 day)
Default value: 30
Example: scio const -s s0:syndef set sc_syndef_report_freq 60

sc_syndef_log_detail
Description: Controls whether or not the destination IP address appears in the log’s variable data.
Possible values: 0-1 (0 = off, 1 = on)
Default value: 1
Example: scio const -s s0:syndef set sc_syndef_log_detail O

sc_syndef_log_ports

Description: Controls whether or not the destination port appears in the log's variable data. If both sc_syndef_log_detail and sc_syndef_log_ports are set to 1 (on), the sc_syndef_log_ports value takes precedence and is displayed, not the IP.
Possible values: 0-1 (0 = off, 1 = on)
Default value: 0
Example: scio const -s s0:syndef set sc_syndef_log_ports 1

getsystem

Use the getsystem command to display Sensor system information.

Syntax	`scio getsystem`	
Options	Function	Arguments
`None`	Displays Sensor system information.	None

Results appear as follows:

Product Name: IDP-xxxx (e.g. IDP-200, IDP-600c, IDP-1100f)
Serial Number: XXXXXXXXXXXXXXXXXXXX
Software Version: major.minor.build_number (libsc.o)
IDP Mode: sniff|router|proxyarp|bridge|transparent
HA Mode: Enabled|Disabled
Detector Version: major.minor.build_number

ha

Use the ha command to manage high availability (HA) clusters.

Syntax	`scio ha options arguments`	
Options	Function	Arguments
define	Defines a new HA cluster.	< ha cluster > < ha id > < subs name >
undef	Clears an HA cluster definition.	< ha cluster >
use	Configures an HA cluster to use a virtual circuit as an HA interface.	< vc name > < ha cluster >
unuse	Configures an HA cluster to stop using a virtual circuit as an HA interface.	< vc name > < ha cluster >
status	Displays the status of a Sensor.	< sub name >
loadbalance status	Displays the status of standalone HA nodes.	None

nic

Use the nic command to attach or detach the kernel module to or from a network interface card (NIC). *Default setting:* Attach/detach all NICs.

Syntax	`scio nic options arguments`	
Options	Function	Arguments
`attach`	Attaches kernel module to a NIC.	< nic name >
`release`	Detaches a kernel module from a NIC.	< nic name >

policy

Use the policy command to manually load and unload a security policy, look up rules in a policy, and verify a policy and detector engine.

If Management Server flows are not exempted from IDP detection methods, security policies that drop any traffic for any service can create communication problems between the NSM server and the UI. By default, the Sensor exempts all flows between the Sensor and the IDP Management Server and does not apply security policy rules to Sensor-Management Server connections.

To view or modify this setting, edit the general parameters in the Sensor Settings rulebase in the UI.

Syntax	`scio policy options arguments`	
Options	Function	Arguments
load	Installs a policy and detector engine on a Sensor.	< subscriber > < filename > < detector >
unload	Uninstalls a loaded rulebase.	< subscriber > < filename > < rulebase > < sip > < dip > < svc > < detector >
lookup	Displays the matching rules from a rulebase in a policy file.	< subscriber > < rulebase > < filename >
verify	Verifies a policy and detector engine.	< subscriber > < filename > < detector >

ssl

Use the ssl command to manage RSA keys for SSL Inspection.

Each Sensor supports up to 100 server private keys and up to 100 servers per key.

NOTE: Other SSL-related commands are listed under the const heading below.

Syntax	`scio ssl options arguments`	
Options	Function	Arguments
list all	Lists all registered key IDs and associated SSL servers (server ID and IP address).	None
list key	List all servers associated with a particular key.	< key ID >
add key	Add a new RSA private key. Key may optionally be assigned to a server when created. A key ID (0-99) is returned. Encrypted keys require a password.	< key path > [password < password string >] [server < server-ip >]
add server	Add a server to a particular key.	< server IP > key < key ID >
delete all	Remove all registered keys and servers.	None
delete key	If just a key is specified, remove a particular key and all its associated servers. If a key and server are specified, removes the server but not the key.	< key ID > [server < server IP >]

Configuring SSL Inspection

To turn on SSL inspection:

1. Use SCP or FTP to copy your private key file to the Sensor.

   The Sensor does not run an FTP server, so you have to initiate the FTP session from the Sensor.

2. Add the key to the Sensor keystore. You can add a server with this step or wait until the next step.

   To add a key that doesn't have a password:

   scio ssl add key <key path > server <server IP>

   To add a key with a password:

   scio ssl add key <key path> password <password> server <server IP>

3. Retrieve the key ID from the keystore:

   scio ssl list all

4. Add any other servers that use the same key:

   scio ssl add server <server IP> key <key ID>

5. Turn on SSL Decryption:

   ```
   scio const -s s0 set sc_ssl_decryption 1
   ```

The Sensor now inspects traffic for which it has a key/server pair.

Table 21: Supported Encryption Algorithms

Cipher	Exportable	Type	Key Material	Expanded Key Material	Effective Key Bits	IV Size
NULL	No	Stream	0	0	0	N/A
IDEA_CBC	No	Block	16	16	128	8
RC4_128	No	Stream	16	16	128	N/A
DES_CBC	No	Block	8	8	56	8
3DES_EDE_CBC	No	Block	24	24	168	8
AES_128_CBC	No	Block	16	16	128	16
AES_256_CBC	No	Block	32	32	256	16

Table 22: Supported SSL Ciphers

Version	Cipher Suites	Value
TLS/SSLv3	TLS_RSA_WITH_NULL_MD5	0x0001
	TLS_RSA_WITH_NULL_SHA	0x0002
	TLS_RSA_WITH_RC4_128_MD5	0x0004
	TLS_RSA_WITH_RC4_128_SHA	0x0005
	TLS_RSA_WITH_IDEA_CBC_SHA	0x0007
	TLS_RSA_WITH_DES_CBC_SHA	0x0009
	TLS_RSA_WITH_3DES_EDE_CBC_SHA	0x000A
	TLS_RSA_WITH_AES_128_CBC_SHA	0x002F
	TLS_RSA_WITH_AES_256_CBC_SHA	0x0035

subs

Use the subs command to manage Sensors on a virtual circuit.

Syntax	`scio subs options arguments`	
Options	Function	Arguments
define	Defines a new Sensor.	< sub name >
undef	Clears a Sensor definition.	< sub name >
attach	Attaches a Sensor to a virtual circuit.	< sub name > < vc name >
release	Detaches a Sensor from a virtual circuit.	< sub name > < vc name >
qmodules	Displays qmodules for a Sensor.	< sub name >
qmodstats	Displays qmodules statistics for a Sensor.	< sub name >
status	Displays the status of the Sensor.	< sub name >
list	Displays all defined Sensors.	None
rulestats	Displays the rule statistics for the Sensor.	< sub name >
reset	Resets the statistics that the subscriber maintains (packets, flows, peak throughput, etc.)	< sub name >

sysconf

Use the sysconf command to display Sensor configuration information.

Syntax	`scio sysconf options`
Options	Function
protocols	Displays protocols supported by the Sensor.
attacks	Display analyzed attacks supported by the Sensor.
contexts	Displays contexts that can be searched for attacks.
ptypes	Displays protocols the kernel can detect.
all	Displays all statistics shown above for a Sensor.

var

Use the var command to display variable settings for a Sensor or virtual router.

Syntax	`scio var options arguments`	
Options	Function	Arguments
-v	Displays variables for a virtual router.	< vr name > < varname >
-s	Displays variables for a Sensor.	< subscriber > < varname >

Examples:

```
scio var -s s0 sc_tcp_flow_table
scio var -v vr0 sc_arp_table
```

vc

Use the vc command to manage virtual circuits and Sniffer mode.

Syntax	`scio vc options arguments`	
Options	Function	Arguments
list	Displays defined virtual circuits.	None
define	Specifies a virtual circuit.	< vc name >
undef	Clears a virtual circuit definition.	< vc name >
external	Sets or unsets an interface as an external interface. Logs generated from packets arriving on this interface have the external bit set or unset.	< vc name > < set | unset >
sniff	Enables or disables Sniffer mode.	< vc name > enable < vc name > disable

version

Use the version command to show the version of the Sensor.

Syntax	`scio version`	
Options	Function	Arguments
None	Displays version of the Sensor.	None

vr

Use the vr command to manage virtual routers.

NOTE: The virtual router on the Sensor is actually a virtual path, a logical grouping of the Sensor's network interfaces creating a logical circuit for a network segment's traffic traveling through the IDP. The IDP virtual path does not support the same functionality as a virtual router on a Juniper Networks firewall/VPN device.

Syntax	`scio vr options arguments`	
Options	Function	Arguments
list	Displays all defined virtual routers.	None
define	Defines a new virtual router.	< vr name >
undef	Clears a virtual router definition.	< vr name >
attach	Attaches a virtual router to a virtual circuit.	< vr name > < vc name >
release	Detaches a virtual router from a virtual circuit.	< vr name > < vc name >
mode	Enables Bridge, router, or Proxy mode. Shows current mode.	router < vc name > bridge < vc name > enableswitch < vc name >
addmac	Adds a multicast address to a virtual router.	< m address > < vr name > < vc name >
delmac	Deletes a multicast address on a virtual circuit.	< m address > < vr name > < vc name >
listmac	Displays multicast addresses for a virtual router.	< vr name >
addstaticmac	Adds a MAC address to a MAC table.	< vr name > < mac address > < vc name >
delstaticmac	Deletes a MAC address from a MAC table.	< vr name > < mac address >
addarp	Adds an ARP entry to a virtual router.	< vr name > < ip address > < mac address > < vc name >
delarp	Deletes an ARP entry from a virtual router.	< vr name > < ip address >
showspan	Displays Spanning Tree Protocol (STP).	None

sctop Commands

Use sctop commands to monitor the Sensor connection tables and view Sensor status:

Location	/usr/bin
Syntax	sctop options
Options	Function
-h	Displays help for the sctop utility.
-a	Displays the ARP/MAC table.
-i	Displays the IP flows.
-c	Displays the ICMP flows.
-u	Displays the UDP flows.
-t	Displays the TCP flows.
-r	Displays the RPC program table.
-x	Displays the RPC XID table.
-s	Displays status information about the Sensor.
-m	Displays system memory statistics.
-l	Displays Q-module statistics.
-e	Displays rulebase statistics.
-g	Displays aggregate statistics.
-k	Displays attack statistics.
-p	Displays Spanning Tree Protocol (STP) information.
-b	Displays IP Action table.
-z	Displays packet distribution.
-d	Displays the strip chart, a text-based chart for packet/second, kbits/second, and the sessions that the UI sees.
-f	Displays fragment chain.
-w	Displays HA status.
-y	Displays IDS cache statistics.
-v	Sorts in reverse order.
-0	Disables sorting.
-1	Sorts by bytes per session.
-2	Sorts by packets per session.
-3	Sorts by expiration.
-4	Sort by service.
-5	Sorts by destination port.
-6	Sorts by source address.
-7	Sorts by destination address.

Example: SCTOP Flow Table

Source-IP	Port	Destination-IP	Port	Flag	Dir	State	Service	Timeout
10.150.98.62	4137	10.150.20.43	139	R----	-> >	Ltn	SMB	30/30
10.150.20.43	139	10.150.98.62	4137	R----	< <-	Close	-	30/30
10.150.73.39	6000	10.150.20.242	43117	R----	-> >	Ltn	-	30/30

The Flag column has five sections, with the following options:

- Flow State		- Management Flow		- Auxiliary Flow		- Packet Logging		- Flow Sync	
R	Ready	m	Management Flow	a	Auxiliary Flow	p	Packet Logging	h	Flow failed over
A	Anticipated								
V	Virtual	-	Non-Management Flow	-	Non-Auxiliary Flow	-	No Packet Logging	-	Normal Flow
X	Rejected							s	Flow synced from another IDP
U	Unknown								

NOTE: The flow state “Unknown” should not appear.

Flag examples:

- R----

  Flow is Ready, nonmanagement, nonauxiliary, no packet logging, normal.

- Rm---

  Flow is Ready, management, nonauxiliary, no packet logging, normal.

- A--ps

  Flow is Anticipated, nonmanagement, nonauxiliary, with packet logging, and synched over from another IDP.

bypassStatus Command

Pairs of copper forwarding interfaces on the IDP 50, 200, 600, and 1100 appliances have a built-in bypass feature (NIC Bypass). When enabled, the bypass feature causes the two ports to physically connect if the Sensor stops responding, letting traffic continue to flow through the device.

NIC Bypass is only available when the Sensor is in Transparent mode.

Fiber interfaces must use an external bypass device.

NIC Bypass works using a watchdog timer. The Sensor sends each timer a signal every second telling it to reset. If the timer does not receive a reset signal for three seconds, bypass activates. If the timer then receives a reset signal, bypass deactivates and the Sensor goes back to normal operation.

The bypassStatus command displays the state of the bypass daemon, the watchdog timer setting, the watchdog timer reset interval, and the bypass state of each copper pair.

Location	`/usr/idp/device/utils/`	
Syntax	`bypassStatus` *interval iterations*	
Options	Function	Arguments
	If no options are specified, the status displays once.	None
interval	The time, in seconds, between status displays. If an interval is specified, the status will display every *interval* seconds until Ctrl + C is pressed.	None
`iterations`	The number of times the status is displayed, if an interval has been specified. If both *interval* and *iterations* are specified, the status will display every *interval* seconds, *iterations* times.	None

statview Command

The statview command lets you view information stored in your Application Volumen Tracking tables.

To turn on statistical profiling, set the sc_periodic_stat_update constant to 1. See Application Volume Tracking Constants on page 181.

For more information on using Application Volume Tracking, see Application Volume Tracking on page 43.

Location	`/usr/idp/device/utils/`	
Syntax	`statview` *-d [< db_dir >] option arguments*	
Options	Arguments	Description
meta	None	Display information about the stat table.
view	[-r < stat_file >] | [< fromTime > < interval >]	View aggregate statistics for all rows of all tables that correspond to the indicated time range.
`query`	-w < outfile > -a [ip|proto|port] -r < infile > | [< fromTime > < interval >]	Query a stat table for data collated by IP addresses, protocol, or ports.
`chart`	< interval >	Get statistics data from all tables of type of interval.

< fromTime > is of the format mm:hh:DD:MM:YYYY:(std|dst). < fromTime > must match an interval start as displayed using the **statview meta** command.

< interval > is either 15M or 1H, depending on whether you want to examine a 15-minute interval or a 1-hour interval file.

The < db_dir > and < stat_file > options are only necessary if you have copied the interval files to a different location.

Appendix B
Daemons

Daemons that run on the Sensor handle various processing and monitoring tasks. This appendix provides a summary of these daemons.

Sensor Daemons

IDP Daemons on the Sensor:

Daemon	Purpose
agent	Brings up the secure TLS channel to NetScreen-Security Manager (NSM). Sends Sensor status, logs, and profiled data to NSM. Agent also receives policy, detector, and configuration commands from NSM.
dLogPurger	Purges logs when disk is full or nearly full.
idp	Kernel module that is the core IDP Engine.
idpLogReader	Reads Sensor logs and writes them to local hard disk.
nicBypass	Controls the Internal Bypass feature.
peerPortModulator	Controls peer port modulation functionality.
pkid	Inspects SSL traffic, if SSL inspection is turned on.
profiler	Profiles network and application data collected by the Sensor.
schad	Performs load balancing and failover between Sensors in HA configuration.
sciod	Handles policy push, Sensor information retrieval, Profiler status, and so on.
sessionFetcher	When user wants to display session packet capture in UI, retrieves session data and sends it to the IDP Management Server.
slogd	Logs packet captures to Sensor hard disk.

Appendix C

Common Criteria EAL2 Compliance

This appendix describes actions that are required for a security administrator to properly secure the IDP Sensor to be in compliance with the Common Criteria EAL2 security target. Further Common Criteria information is available in the *NetScreen-Security Manager Administrator's Guide.*

Guidance for Intended Usage

- In order to collect data or to prevent certain data from passing to or from IT systems, the IDP Sensor must be connected to the network from which IT systems are to be monitored.
- The IDP Sensor must be appropriately scalable to the IT system that it monitors.
- The IDP Sensor must be managed in a manner that allows it to appropriately address changes in the IT system that it monitors.
- The IDP Sensor, NetScreen-Security Manager Device and GUI Servers, and the NetScreen-Security Manager UI must be installed on dedicated systems. These dedicated systems must not contain user processes that are not required to operate the IDP system.

Guidance for Personnel

- There must be one or more competent individuals assigned to manage the IDP Sensor, NetScreen-Security Manager, and the security of the information that they contain.
- The authorized administrators must not be careless, willfully negligent, or hostile and must follow and abide by the instructions provided by the IDP Sensor, NetScreen-Security Manager, and UI documentation.
- The IDP Sensor and NetScreen-Security Manager must be accessed only by authorized users.

Guidance for Physical Protection

- The processing resources of the IDP Sensor, NetScreen-Security Manager, and NSM UI must be located within facilities with controlled access which prevents unauthorized physical access.

Appendix D
IVE Signaling Setup

This appendix covers the following topics:

IVE Signaling Feature Overview

IVE Signaling allows the IDP Sensor to notify the IVE appliance of security events based on parameters provided by the IVE administrator. The IVE appliance can then take action against the offending connection based on information in the log entry sent by the IDP.

The IDP administrator must provide information to the IVE administrator so the two devices can establish communication. After communications have been established, all configuration is done on the IVE side.

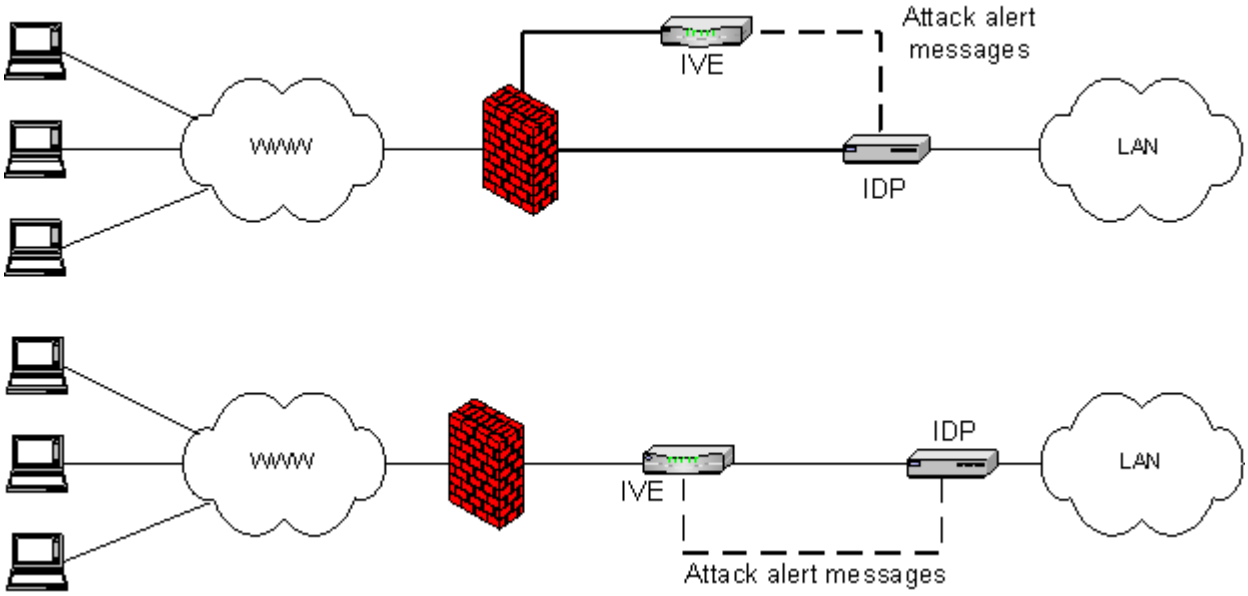
The two appliances work together as follows:

- The IVE appliance tells the IDP Sensor what IP addresses it is interested in.
- The IVE appliance tells the IDP Sensor what event severities it is interested in. For example, the IVE administrator may only want to know about events of Critical severity, or only events of Medium severity or higher.
- When the Sensor detects an event of sufficient severity from an IP address within the pool, it sends a copy of the resulting log entry back to the IVE appliance.
- The IVE appliance then compares the log entry to a set of established rules and takes the action indicated by the administrator during setup.

Possible IVE-IDP Topologies

VPN/SSL traffic must flow through the IVE appliance before it flows to the IDP Sensor. However, the two need not be directly connected inline. For example, the IVE may be in the DMZ while the IDP Sensor may be in the corporate LAN.

Figure 24: IVE-IDP Topologies



IDP Configuration Requirements

Most of the IVE-IDP Signaling configuration is done on the IVE side. However, the IDP administrator must do some configuration on the IDP side.

NOTE: An IDP Sensor can send logs to one IVE appliance only. However, an IVE appliance can receive logs from more than one IDP Sensor.

- Rules in each policy on the Sensor must have `logging` turned on. Since the event notification and subsequent action depends on log entries, a log entry must be generated by the event for the feature to work. Not all rules must have logging turned on, as long as each event generates at least one log entry.

  To turn on logging for a policy:

  1. Select `Security Policies > <policy name>` in NSM.
  2. Select the `IDP` tab.
  3. In the Notification cell for each rule, do the following:
     a. Right-click and select `Configure`.
     b. Check the `Logging` checkbox.
     c. Click `OK`.
     d. Repeat for each rule in the policy.
  4. Push the updated policy to the Sensor.

- Each Sensor must have `log suppression` turned off.

  The log suppression features suppresses multiple similar entries from the same source IP address. Numerous virtual connections coming through the IVE appliance may appear as only one IP address to the IDP. Therefore, log suppression must be turned off so that the Sensor informs the IVE of every detected event.

  To turn off log suppression for a single Sensor:

  1. Select `Device Manager > Security Devices` in NSM.
  2. Double-click the Sensor you want to modify.

     The Device Editor displays.
  3. Select `Sensor Settings`.
  4. Uncheck the `Enable log suppression` checkbox.
  5. Click OK to save your changes.
  6. Right-click your Sensor and select Update Device.

  To turn off log suppression multiple Sensors using a template:

  1. Select `Device Manager > Security Device Templates` in NSM.
  2. Double-click the template you want to modify, or click the + button to create a new template.

     The Device Template Editor displays.
  3. If you are creating a new template, enter a template name in the Name field.
  4. Select `Security > IDP SM Settings`.
  5. Uncheck the `Enable log suppression` checkbox.
  6. Click OK to save your changes.
  7. Select Devices > Configuration > Template Operations from the NSM menu bar.
  8. Use the Template Operations dialog to assign the template to Sensors. Refer to the *NetScreen-Security Manager Adminitrator's Guide* for more information on the Template Operations dialog.

- IP Actions (such as IP Block and IP Close) should be use with extreme care, if at all, in policies that examine traffic from an IVE appliance. Closing or blocking a connection based on IP address may shut down numerous VPN sessions.

  Session-based actions are recommended instead.

See the IDP Concepts & Examples Guide, Chapter 7, for information on setting actions for rules.

Setting Up Signaling

Most of the signaling configuration must be done on the IVE side. However, the IDP administrator must provide certain information to the IVE administrator.

Communication between the IVE appliance and the IDP Sensor is initiated by the IVE appliance. Instructions for setting up communications can be found in the *Juniper Networks SA 2000/SA 4000/SA 6000, NetScreen SA 1000/SA 3000/SA 5000, NetScreen SA FIPS Administration Guide.*

The IVE administrator needs the following information from the IDP administrator:

- For all Sensor modes except router mode, the IVE administrator needs the IP address of the IDP Sensor's Management Port (MGT). For Router mode, the IVE administrator needs the IP address of a forwarding interface.
- The Sensor's IVE one-time password (OTP). Instructions for generating the IVE OTP are below.

Generating the IVE OTP

The IDP administrator generates a one-time password (OTP) for establishing communication with the IVE appliance. The IDP administrator gives this OTP, as well as the IP address of the Sensor, to the IVE administrator.

To generate an IVE OTP:

1. Open the ACM on the Sensor.
2. Click the `Reconfigure Management Server and IDP IVE Communication` link.
3. Check the `Reset IVE OPT?` checkbox.
4. Click `Next Step`.

   A new OTP is generated and displayed in the IVE OTP field of the Final Configuration Report.
5. Click `Confirm Configuration` to save the new IVE OTP.
6. Give the IVE OTP to the IVE administrator.

Once communication between the devices has been established, configuration is handled on the IVE side. The IDP Sensor provides information to the IVE appliance based on the IVE configuration.

Appendix E
Glossary

admin user An IDP user type who has IDP Manager access in the NSM UI.

alert 1. A column in the Log Viewer.

2. A log record that represents an important event. If an alert is configured for a rule in the security policy, when the rule is matched the corresponding log record displays an alert in the alert column of the Log Viewer. Security administrators use alerts to become aware of and react to important security events.

3. Warnings that correspond to the performance of IDP devices or device processes. You configure alerts in the Device Monitor.

Appliance Configuration Manager (ACM) The web-based interface for configuring the IDP appliance.

attack Attacks attempt to exploit vulnerabilities in computer hardware and software. Depending on the severity of the attack, it might disable your system completely, allow an attacker to gain confidential information stored on your system, or use your network to attack other networks.

attack object A signature or protocol anomaly that is combined with context information. Attack objects are used in Main rulebase rules to match malicious traffic patterns. Each attack object detects a known attack or protocol anomaly that can be used by an attacker to compromise your network.

attack summary A section of the Dashboard that displays the top 10 attacks on a network over a specified period.

backdoor A mechanism installed on a host computer that facilitates unauthorized access to the system. Attackers who have already compromised a system can install a backdoor to make future attacks easier. When attackers send and retrieve information from backdoor, they generate interactive traffic, which can be detected and prevented by the Backdoor Detection rulebase.

Backdoor Detection rulebase A rulebase in the Security Policy Editor used to protect a network from backdoor attacks by detecting and preventing unauthorized interactive traffic.

binding The connection that links a signature to the service context in which it occurs. The service binding for each signature is defined in the signature attack object. You can view or set bindings in the Signature Editor dialog box.

Bridge mode A deployment mode of the IDP system. In this mode, the IDP system acts as a networking bridge, forwarding packets within the same logical network subnet, which is broken into several physical parts. Packets are forwarded transparently; you do not need to configure other network devices to be aware of the IDP Sensor.

buffer overflow An event that occurs when a program or process attempts to store more data in a buffer than the buffer was intended to hold. Buffers, which provide temporary data storage, are designed to contain a finite amount of data; any additional data can overflow the buffer zone and attempt to enter nearby buffers, corrupting or overwriting that buffer's existing data.

BugTraq A moderated mailing list that discusses and announces computer security vulnerabilities. www.bugtraq.com.

cluster A group of two to sixteen IDP Sensors that provides HA for failover protection. The IDP system handles the HA cluster as a single device; security policies that are installed to the HA cluster are applied to all IDP Sensors within the HA cluster.

command line utilities A set of utilities that provides management and troubleshooting functionality from the IDP system command line. Command line utilities include scio (configures the IDP system), sctop (monitors the connection status and views Sensor status), and idp.sh (starts, stops, and monitors Sensor processes). IDP Sensor command line utilities are located under the /usr/idp/device/bin directory and the /usr/idp/device/utils directory.

Common Vulnerabilities and Exposures (CVE) forum Standardized list of vulnerabilities and other information security exposures.

console port (CONSOLE) A DB-9 serial port on the IDP Sensor. This port can be used to access the Sensor Command Line Interface (CLI).

context The matching criteria for an attack object: line, packet, stream, or network traffic protocol. Specifies where IDP should look for patterns to match attack objects.

Dashboard A customizable UI component that provides vital real-time statistics for your network.

dependent axis The axis of the Log Investigator that provides secondary information based on the primary information in the independent axis. Example: If the dependent axis is set to Top Sources and the independent axis is set to Top Attacks, the Log Investigator displays the most popular source address and the attacks that each source generated.

device monitor A component of the UI that displays real-time availability and performance status information on IDP Sensors and NSM on your network.

disabled rule A rule that has been temporarily removed from inclusion (but not deleted) in a security policy rulebase.

distributed port scan An attack that uses multiple source addresses to scan ports on your network. Distributed port scans can occur over multiple connections and sessions and can be detected and prevented in the Traffic Anomaly rulebase.

Domain Name System (DNS) Allows you to enter a host name instead of an IP address for network objects. If DNS is configured, the Sensor can also resolve IP addresses by performing domain-name lookups from the Sensor console.

false positive Any situation in which benign traffic causes an IDS to generate an alert; also known as a *false alert*.

filter A method of sifting log records by individual Log Viewer columns.

flag A setting applied to log records in the Log Viewer. A flag setting can be one of the following: High, Medium, Low, Closed, False Positive, Assigned, Investigate, Follow-Up, or Pending.

flow hash table A table that contains traffic-flow information.

flow tracking A method of reducing false positives. Flow tracking correlates multiple TCP or UDP connections into a single flow to determine the validity of the traffic.

forwarding interface An Ethernet port (interface) that the IDP Sensor uses to forward traffic to the external or internal network.

Fully Qualified Domain Name (FQDN) Consists of a host and domain name, including the top-level domain. For example, www.example.com is a fully qualified domain name: www is the host, mycompany is the second-level domain, and .com is the top level domain. A FQDN starts with a host name and continues to the top-level domain name, so www.sensor1.example.com is also an FQDN.

group An object in the IDP system that contains related objects. You can create network, service, or attack object groups to use in security policies or Dashboard watch lists just as you would individual objects.

Heartbeat (HB) protocol The protocol that determines how heartbeats operate. HB protocol sends one or more heartbeats to all other IDP Sensors in the HA cluster once every interval. If an IDP Sensor fails to send a heartbeat for the specified number of times in a row (the failure count), the other IDP Sensors consider that Sensor inactive and begin to accept traffic for that Sensor.

heartbeat interface The interface that the IDP Sensor uses to send heartbeats. IDP Sensors joined in an HA configuration use heartbeats to share state information with each other. Heartbeats use the Heartbeat (HB) protocol.

high availability (HA) The ability to provide uninterrupted service if a component of the system fails. IDP supports three HA solutions: load balancing, hot standby, and Spanning Tree Protocol (STP).

Host Watch List A configurable section of the Dashboard that displays the important hosts for a specified network.

hot standby An HA configuration in which a primary IDP Sensor handles all network traffic while a secondary IDP Sensor stands by. If the primary IDP Sensor fails, network traffic is redirected to the secondary IDP Sensor.

ICMP sweep An attack that uses a single source address to ping multiple IP addresses on your network.

Internet Control Message Protocol (ICMP) A protocol that allows the passing of error, control, and informational messages.

IDP appliance The hardware device that contains the IDP Sensor software preinstalled. You configure the IDP Sensor software on the IDP appliance when you install the IDP system.

IDP Sensor One of the three tiers of the IDP system. IDP Sensors can operate as passive sniffers to monitor network traffic or as in-line devices that detect and protect against intrusions by sending alerts and resetting or dropping connections. The Sensor software is preinstalled on the IDP appliance and can be configured to protect your networks and hosts using the Appliance Configuration Manager (ACM).

idp.sh A command line utility that starts, stops, and monitors the status of IDP Sensor processes.

independent axis The axis of the Log Investigator that defines the primary set of information. For example, if the independent axis is set to Top Attacks and the dependent axis is set to Top Sources, the Log Investigator displays the most popular attacks and the source addresses that generated them.

informational signature / event Normal, harmless traffic, containing URLs, DNS lookup failures, and SNMP public community strings. Informational attack objects are designed to match such traffic to obtain information about your network.

initialization state A state of the IDP Sensor. When a Sensor initially powers on, it enters the initialization state. When initialization of the Sensor is complete, it moves to an active state and begins forwarding traffic. You can also choose to forward traffic during the initialization state.

in-line The position of the IDP appliance in the network traffic flow. When IDP is deployed in-line, it is positioned directly in the path of packets, where it can actively prevent attacks.

interactive traffic Network traffic that indicates human involvement in a normally automated process, such as a user typing commands. Interactive traffic looks different from other traffic because humans are manually controlling one end of the connection. In a connection between two programs, the data transfer is automated; TCP packets can batched and sent in bulk for efficiency. In a connection between a program and a user, packets are sent when they become available; characters display as they are typed (not after the word is complete). Interactive programs transmit several short IP packets containing individual keystrokes and their echoes, reflecting the real-time actions of a user (or attacker).

Internal Bypass Copper forwarding interface pairs in the IDP 50, 200, 600, and 1100 appliances have this feature. If the Sensor is set for Transparent mode, and Internal Bypass is turned on, then the pair of ports will fail over to a Passthrough mode if the Sensor fails. Traffic continues to flow through the network.

IP action Traffic-handling instructions for the IDP Sensor. IP actions are dynamically generated actions (triggered by rule match) that can affect future traffic based on previous traffic. You can set IP actions in the Main, Network Honeypot, or Traffic Anomalies rulebases.

IP defragmentation and TCP reassembly A method of reducing false positives. IP defragmentation reconstructs fragmented traffic.

IP spoofing The practice of mimicking, or *spoofing*, the source address of an IP packet. Every IP packet includes the destination address (where the packet is going) and the source address (where the packet came from). The routers that provide Internet communication between distant computers determine the best route for the IP packet using only the destination address and typically ignore the source address. An attacker can fake the source address of a malicious IP packet (by modifying the packet headers) so that the packet appears to come from a trusted system.

key A unique name that identifies an attack object internally in the IDP system and the Log Viewer.

link-check An activity that verifies the status of the IDP appliance's interfaces. Each interface pings specified IP addresses at regular intervals to ensure that they can send and receive traffic.

load balancing An HA configuration in which all IDP Sensors in an HA cluster share network traffic equally. If one IDP Sensor fails, network traffic is redirected to the other IDP Sensors in the HA cluster.

log A grouping of log records.

log record A record of a security event. A log record is created when IDP rules are triggered or when the IDP system performs a standard function. Log records are stored in log files in NSM.

Log Viewer The UI component used to view log records.

main display area The area of the UI where components are displayed.

Main rulebase A rulebase in the Security Policy Editor that protects the network from attacks by using signature and protocol-anomaly attack objects to detect and prevent malicious activity.

management interface (MGT) The IDP appliance Ethernet port (interface) that is used for communication between the IDP Sensor and NSM. Each IDP Sensor has a management interface with an IP address that must be accessible to NSM.

MD5 checksum A verification mechanism that provides data integrity by hashing and sequencing the data to ensure that it has not been altered or stolen as it was transmitted over a network. In an HA cluster, a user-defined shared secret is used to calculate the MD5 checksum of the heartbeat.

Media Access Control (MAC) Physical hardware addresses that uniquely identify each node of a network. The MAC address is assigned by the hardware manufacturer when the hardware is produced.

menu bar The area of the UI that contains clickable commands. You can access many menu-bar commands using keyboard shortcuts.

Multi-Method Detection (MMD) A combination of detection methods. Each rulebase in the IDP system uses a specific detection method to identify and prevent attacks. Together, these detection methods provide an MMD system that can thwart almost any attack, including reconnaissance, network-level, application-level, and backdoor intrusion attempts.

navigation tree The area of the UI that lists the UI components. When you select a component in the navigation tree, the component is displayed in the main display area.

netmask A string of zeroes (0) and ones (1) that screen, or mask, the network section of an IP address so that only the host computer section of the address remains. Netmasks are used to produce ranges of IP addresses from a single IP address.

network Two or more computers and devices that are connected to each other. A connected computer or device is called a *network component*; components can share the information and resources of other components on a network.

Network Honeypot rulebase A rulebase in the Security Policy Editor that protects the network by impersonating open ports. The Network Honeypot rulebase can detect and prevent port scans and other information-gathering activities.

network object A representation in the IDP system of a workstation, a router, a switch, a subnetwork, an IDP Sensor, or another device on your network. Network objects represent your network topology in the Object Editor, security policies, and log records.

network scan An attack in which a single source address attempts to connect to multiple IP addresses on a network. Network scans provide attackers with valuable information about your network configuration.

Network Time Protocol (NTP) A time-synchronization system for computer clocks that operates through the Internet.

object The building block of the IDP system. Network objects represent components of your network; service objects represent services running on your network; and attack objects represent specific patterns of malicious activity or protocol anomalies within a connection.

Offline mode A Log Viewer mode in which new log records do not appear. The IDP Sensor continues to generate and store log records in Offline mode but does not display them. To see new log records, you must select to view the Log Viewer in Online mode.

One-Time Password (OTP) The OTP seeds the encryption between the Sensor and NSM. You use the OTP when you add a Sensor to NSM.

online mode A Log Viewer mode in which new log records appear as soon as they are generated by the IDP Sensor and transferred to NSM.

Packet Viewer The Log Viewer window that displays the data from an individual packet in your network traffic that corresponds to a log record. Packet data is available in a log record if the matching rule is configured to log packets.

Passive Sniffer mode A deployment mode of the IDP system. When IDP is deployed as a passive sniffer, it monitors and logs network traffic only and cannot provide attack prevention functionality.

pattern A regular expression that is used as matching criteria in an attack object.

port scan An attack in which a single source address attempts to connect to every port on a single machine. Ports scans provide attackers with valuable information about your network configuration.

process status A view of the Device Monitor that displays availability and performance status information for processes on IDP Sensors on your network.

protocol anomaly A deviation from the RFC specifications that dictate how communications between two entities should be implemented. Most legitimate traffic does not deviate from the protocols; when anomalies are detected they are often a sign of malicious traffic and seen as a threat to the system.

protocol-anomaly detection A detection method, also called *protocol analysis*, that identifies protocol anomalies. Protocol-anomaly detection determines illegal or ambiguous packets that can constitute security threats by checking them against the protocol RFCs or the definitions imposed by the network administrator.

protocol normalization A method of reducing false positives. Normalizes traffic into a common format for accurate analysis.

proxy-ARP loop A loop that occurs when the IDP Sensor answers an ARP request for an IP address owned by another IDP appliance in the HA cluster. To prevent proxy-ARP loops, you must configure each Sensor in the HA cluster to ignore the forwarding interfaces of the other IDP appliances.

Proxy-ARP mode A deployment mode of the IDP system. In this mode, the IDP system acts as a proxy, sending ARP requests and replies between networks. An ARP request/reply is a mechanism for resolving IP addresses. Network devices coming online send out ARP requests to determine if a particular IP address is being used. With IDP, network nodes use the Sensor's MAC address to send network traffic across segments. The IDP Sensor relays existing ARP requests between networks and proxies replies by issuing its own ARP replies. If the IDP Sensor is inserted between network segments, network nodes attached to these segments might need to update their cached ARP entries.

read-only user An IDP user privilege type that has limited command access to the UI. Read-only users cannot take control of or write to NSM.

read-write user An IDP privilege user type that has elevated command access to the UI. Read-write users can write to NSM.

Remote Procedure Call (RPC) A type of protocol that allows a program on one computer to execute a program on a server computer.

reports 1. Summarize threats and traffic behavior over time based on filtered log records. 2. A section of the Dashboard that displays two selected IDP reports.

Reports component The UI component displays reports. You can generate and view reports based on filtered log records that summarize threats and traffic behavior over time.

Request for Comments (RFC) extension An addition to an RFC document that extends the regulatory domain of the original RFC and defines additional requirements. An RFC extension can also clarify ambiguous or implied requirements of the original RFC.

root user account (Sensor) A UNIX user account that is used to log in remotely to the IDP Sensor.

Router mode A deployment mode of the IDP system. In this mode, the IDP system acts as a traditional IP router: the Sensor accepts packets from an attached network, examines the destination address, consults its routing tables, and forwards the packets accordingly.

routing table A table that contains network route information. The IDP Sensor's routing table contains route information about the path that data packets should use when traveling through the IDP Sensor. When a new data packet enters the network, routing algorithms look at the packet's source and determine the optimal route to the packet's destination. This information is stored in the routing table; other data packets with the same source and destination automatically use the route specified in the routing table.

rule A user-defined match/action sequence. Rules are represented graphically in the Security Policy Editor, where you can create, modify, delete, and reorder them in a rulebase.

rulebase A set of rules that uses a specific detection mechanism to identify and prevent attacks.

schad daemon A process that runs on each IDP Sensor in the HA cluster. The schad daemon provides a method for the Sensor to determine its state, sends heartbeats to other Sensors in the HA cluster, and passes cluster state information to the Sensor's kernel module.

scio A command line utility that you can use to configure the IDP system.

sctop A command line utility that you can use to monitor connection tables and display IDP Sensor status.

Secure Shell (SSH) Provides secure remote access to devices. You can use SSH to access the IDP Sensor remotely.

security policy A set of rulebases. Security policies are created in the UI and stored in NSM so they can be accessed from a remote client anywhere on the IDP-protected network. Security policies are installed on IDP Sensors, which log network traffic based on the rules within the security policy.

Security Policy Editor The UI component where you create and manage security policies.

Sensor Settings rulebase A rulebase in the Security Policy Editor. You can use the Sensor Settings rulebase to fine-tune the performance of an IDP Sensor to your network traffic.

service An Application Layer protocol that specifies how communication between two systems (applications, servers, Ethernet cards, and so on) occurs.

service object A representation of a service that is supported on your network.

session A period of time over which an event is performed.

severity The designated threat level of an attack (critical, high, medium, low, or informational). Attack objects use the severity setting that matches the threat level of the attack they detect.

shared secret A user-defined alphanumeric text string that authenticates heartbeat messages and verifies that the heartbeat has not been altered during transmission.

signature A pattern that exists within an attack. The signature pattern can be a specific segment of code, a URL, a value in a packet header, and so on. Signatures are used to create signature attack objects, which use the signature pattern to detect the attack and prevent it.

Simple Network Management Protocol (SNMP) Governs network management and the monitoring of network devices and their functions.

sniffer / gateway interface A user-designated Ethernet port (interface) on the IDP appliance. The sniffer interface sniffs the network traffic as it passes through the network hub or switch. Only IDP Sensors running in Sniffer mode have a sniffer interface.

Sniffer mode A deployment mode of the IDP system. When IDP is deployed as a passive sniffer, it monitors and logs network traffic only and cannot provide attack-prevention functionality.

Source Watch List A configurable section of the Dashboard. Displays the source addresses you want to track because they are suspected or known sources of attacks on your network.

Spanning Tree Protocol (STP) An 802.1d specification by the Institute of Electrical and Electronic Engineers (IEEE). The spanning tree algorithm (STA) determines the best communication path between a switch and a node. All other paths are blocked, but not disabled; if the chosen path becomes unavailable, the algorithm recalculates a new communication path.

stateful signature detection A method of attack detection that uses stateful signatures. A stateful signature knows the pattern it is attempting to find and where to look for that pattern. Stateful signatures produce very few false positives because they understand the context of the attack and can eliminate huge sections of network traffic they know the attack would not be in. Stateful signatures are much smarter than regular signatures: they know the protocol or service used to perpetrate the attack, they know the direction and flow of the attack, and they know the context in which the attack occurs.

state-sync interface (HA port) The Ethernet port (interface) that is used for communication between the IDP Sensors in an HA cluster. All Sensors have a state-sync interface with an IP address that is accessible by all other Sensors in the cluster.

status bar The area of the UI that displays additional information for selected components.

Stealth mode A mode for the forwarding interface in Bridge mode. Stealth mode interfaces do not have IP addresses, which makes them more secure and easier to set up.

synchronization interface The state-sync interface of an IDP appliance in an HA cluster.

SYN-Protector rulebase A rulebase in the Security Policy Editor that protects your network from SYN-flood attacks by ensuring the three-way handshake is successfully performed for specified TCP traffic.

syslog A UNIX system-wide program that records system events in a standard format.

TCP scan An attack method that attempts to connect to every TCP port on a single machine. TCP scans provide attackers with valuable information about your network configuration.

terminal rule A rule that terminates the IDP rule-matching algorithm. When a terminal rule is triggered, IDP does not execute subsequent rules in the rulebase.

Terminal rulebase A rulebase in which all rules are terminal. If a rule in a terminal rulebase is matched, IDP does not execute subsequent rules.

toolbar The area of the UI that contains buttons for common tasks. The buttons displayed in the toolbar are determined by the selected component.

traffic-anomaly detection A method of attack detection that identifies intrusion attempts that span multiple sessions, such as port and network scans. Traffic anomaly detection can detect malicious traffic patterns within the overall traffic flow and usually contains frequency and threshold triggers.

Traffic Anomalies rulebase The rulebase in the Security Policy Editor that protects your network from attacks using traffic-flow analysis to detect and prevent port, network, and distributed port scans.

Transaction Control Protocol (TCP) A primary protocol used on the Internet to enable hosts to exchange data streams.

Transparent mode A deployment mode of the IDP system. Transparent mode is an in-line mode, so it can prevent attacks. Its ports are not assigned IP addresses, so other devices do not have to be reconfigured when it is installed.

On the IDP 50, 200, 600, and 1100 appliances, a pair of copper forwarding interfaces in Transparent mode can use the Internal Bypass feature. If this feature is active, the two ports fail to a Connected, Open mode if something happens to the IDP Sensor. In other words, if power fails or the device locks up, the two ports become physically connected, allowing network traffic to continue uninterrupted.

UDP scan An attack method that attempts to connect to every UDP port on a single machine. UDP scans provide attackers with valuable information about your network configuration.

unique identifier A unique number assigned to an IDP Sensor in an HA cluster. The unique identifier enables state-sync communication with other IDP Sensors in an HA cluster. For load balancing, this assignment is arbitrary and is used only to differentiate Sensors in the cluster. For hot standby, the IDP Sensor with the lowest unique identifier is the primary IDP Sensor.

User Datagram Protocol (UDP) A connectionless protocol that provides a communication mechanism for applications.

user interface (UI) The NSM graphical user interface that IDP users use to manage the IDP system, view system status, manage security policies, and view and examine log record data.

variable data Additional data that is contained in a log record and is relevant to a protocol anomaly.

view Another instance of the Log Viewer. You can create custom views with filters and save them in the Log Viewer.

Virtual Network Computing (VNC) A remote display system that displays a desktop environment from anywhere on the Internet.

virtual router (VR) Provides a hardware and software emulation of a physical router. Virtual routers have independent IP routing and forwarding tables. The virtual router on the Sensor is actually a virtual path, a logical grouping of the Sensor's network interfaces creating a logical circuit for a network segment's traffic traveling through the IDP. The IDP virtual path does not support the same functionality as a virtual router on a Juniper Networks firewall/VPN device.

Windows Internet Naming Service (WINS) Determines the IP address of a networked Microsoft Windows computer.

Index

T

U

V