Cisco IOS IPS 5.x Signature Format Support and Usability Enhancements

First Published: November 17, 2006
Last Updated: March 25, 2011

The Cisco IOS Intrusion Prevention System (IPS) acts as an in-line intrusion prevention sensor that scans packets and sessions as they flow through the router to match any Cisco IOS IPS 5.x signature. These signatures are defined in Extensible Markup Language (XML) format and provide the following functionality:

- Automatic signature updates from local servers. Network administrators can either preserve the user's current configuration of signature actions or override the user's current configuration of signature actions with the current IPS configuration.
- Top-level signature categories to classify signatures for easy grouping and tuning. Group-wide parameters, such as signature event action, can be applied to a group through CLI, so the user does not have to modify each individual signature.
- Encrypted (NDA) signature support.
- Direct Download from CCO capability in IOS IPS feature allows an administrator to use the CLI to specify, download and upgrade to new signatures posted for the IOS directly from Cisco.com. An administrator can also configure the router through the CLI to receive future periodic signature downloads automatically to eliminate the manual maintenance efforts and costs of changing or tuning IPS signatures whenever a new update is posted.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Cisco IOS IPS 5.x Signature Format Support and Usability Enhancements" section on page 33.

Use Cisco Feature Navigator to find information about platform support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for Cisco IOS IPS 5.x Signature Format Support and Usability Enhancements

System and Image Requirements for Cisco IOS IPS 5.x

- Cisco IOS IPS signatures.
- Cisco IOS IPS system requirements depend on the type of deployment, the bandwidth requirements, and security requirements. The larger the number of signatures, the larger the amount of memory consumed.
- You must generate a RSA crypto key and load the public signature on your router for signature decryption.

  This following cisco public key configuration can be cut and pasted directly into your router configuration:

```
crypto key pubkey-chain rsa
named-key realm-cisco.pub signature
key-string
30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
00C19E93 A8AF124A D6CC7A24 5097A975 206BE3A2 06FBA13F 6F12CB5B 4E441F16
17E630D5 C02AC252 912BE27F 37FDD9C8 11FC7AF7 DCDD81D9 43CDABC3 6007D128
B199ABCB D34ED0F9 085FADC1 359C189E F30AF10A C0EFB624 7E0764BF 3E53053E
5B2146A9 D7A5EDE3 0298AF03 DED7A5B8 9479039D 20F30663 9AC64B93 C0112A35
FE3F0C87 89BCB7BB 994AE74C FA9E481D F65875D6 85EAF974 6D9CC8E3 F0B08B85
50437722 FFBE85B9 5E4189FF CC189CB9 69C46F9C A84DFBA5 7A0AF99E AD768C36
006CF498 079F88F8 A3B3FB1F 9FB7B3CB 5539E1D1 9693CCBB 551F78D2 892356AE
2F56D826 8918EF3C 80CA4F4D 87BFCA3B BFF668E9 689782A5 CF31CB6E B4B094D3
F3020301 0001
quit
```

Note You can also access the public key configuration at the following URL: http://www.cisco.com/cgi-bin/tablebuild.pl/ios-v5sigup. Ensure that you have your Cisco userid, and password to access this URL.

- You must load one of the following images on your router to install Cisco IOS IPS 5.x: adventerprisek9, advsecurityk9, and advipservicesk9.

Note To check the current system version, use the **show subsys name ips** command.

IPS 4.x uses a version format of 2.xxx.xxx; IPS 5.x uses a version format of 3.xxx.xxx.

Upgrading from Cisco IOS IPS 4.x to Cisco IOS IPS 5.x Signatures

Cisco IOS IPS 5.x format signatures are not backward compatible with Cisco IOS IPS 4.x. You must reconfigure your Cisco IOS IPS features for use with the IPS 5.x signature format command-line interface (CLI) and features.

When reconfiguring Cisco IOS IPS on a router to convert to the 5.x signature format, you must have the following Cisco IOS IPS 4.x information:

- Cisco IOS IPS rule name (which was specified through the **ip ips name** *ips-name* command)
- Interfaces for which the Cisco IOS IPS rule has been applied

To gather this information, issue the **show ip ips configuration** command, which displays a copy of the existing output.

```
Router# show ip ips configuration
Configured SDF Locations:
disk2:my-signatures.sdf
Builtin signatures are enabled but not loaded
Last successful SDF load time: 05:31:54 MST Sep 20 2003
IPS fail closed is disabled
Fastpath ips is enabled
Quick run mode is enabled
Event notification through syslog is enabled
Event notification through SDEE is enabled
Total Active Signatures: 13
Total Inactive Signatures: 0
Signature 50000:0 disable
Signature 50000:1 disable
Signature 50000:2 disable
IPS Rule Configuration
IPS name MYIPS
Interface Configuration
Interface GigabitEthernet0/1
Inbound IPS rule is MYIPS
Outgoing IPS rule is not set
```

Direct Download from CCO capability in IOS IPS Support

The router must have access to Cisco.com in order for IPS signatures to be upgraded directly from Cisco.com. If the router does not have access to Cisco.com, then signature file updates can still be able to be retrieved from a local server.

Restrictions for Cisco IOS IPS 5.x Signature Format Support and Usability Enhancements

Warning **Do not enable all IPS signatures at once. The router may not be able to compile all of the signatures, resulting in high CPU and memory usage, degraded performance, or a system reload.**

Backward Compatibility

Cisco IOS IPS 5.x format signatures are not backward compatible with Cisco IOS IPS 4.x SDFs.

Cisco 870 Series Platform Support

The 870 series platform with Cisco IOS IPS in Cisco IOS Release 12.4(11)T may experience lower performance relative to previous releases (CSCsg57228). The Cisco IOS IPS performance on the 870 series platform is enhanced in a later 12.4(11)T image rebuild.

On the 870 series platform, Cisco IOS IPS is supported only on the adv-ipservices and the adv-enterprise images. Cisco IOS IPS is the same on both images.

Information About Cisco IOS IPS 5.x Signature Format Support and Usability Enhancements

Cisco IOS IPS Overview

The Cisco IOS IPS acts as an in-line intrusion prevention sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When it detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE). The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. The Signature Event Action Processor (SEAP) can dynamically control actions that are to be taken by a signature event on the basis of parameters such as fidelity, severity, or target value rating. These parameters have default values but can also be configured through CLI. When packets in a session match a signature, Cisco IOS IPS can take any of the following actions, as appropriate:

- Send an alarm to a syslog server or a centralized management interface
- Drop the packet
- Reset the connection
- Deny traffic from the source IP address of the attacker for a specified amount of time
- Deny traffic on the connection for which the signature was seen for a specified amount of time

Cisco developed its Cisco IOS software-based intrusion-prevention capabilities and Cisco IOS Firewall with flexibility in mind, so that individual signatures could be disabled in case of false positives. Generally, it is preferable to enable both the firewall and Cisco IOS IPS to support network security policies. However, each of these features may be enabled independently and on different router interfaces.

Cisco IOS IPS Signature Scanning with Lightweight Signatures

The addition of Cisco IOS IPS signature scanning with lightweight signatures in Cisco IOS Release 15.0(1)M is an enhancement to Cisco IOS IPS that allows loading of larger signatures sets, without consuming significant additional memory or reducing the memory consumed by an existing signature set, by loading equivalent lighter-weight signatures. These signatures are referred to as lightweight signatures.

For the signatures made obsolete by new lightweight signatures from new signature scanning engines, Cisco IOS provides the **ip ips inherit-obsolete-tunings** command to apply common parameters the user may have changed (customized) in the old signature file (delta.xml), to the equivalent new signature file. For more information, see the "Enabling Signature Tunings Inheritance" section on page 19.

Signature Categories

Cisco IPS appliances and Cisco IOS IPS with Cisco 5.x format signatures operate with signature categories. All signatures are pregrouped into categories; the categories are hierarchical. An individual signature can belong to more than one category. Top-level categories help to define general types of signatures. Subcategories exist beneath each top-level signature category. (For a list of supported top-level categories, use your router CLI help (**?**).)

Router Configuration Files and Signature Event Action Processor (SEAP)

As of Cisco IOS Release 12.4(11)T, SDFs are no longer used by Cisco IOS IPS. Instead, routers access signature definition information through a directory that contains three configuration files—the default configuration, the delta configuration, and the SEAP configuration. Cisco IOS accesses this directory through the **ip ips config location** command.

Note You must issue the **ip ips config location** command; otherwise, the configuration files are not saved to any location.

SEAP is the control unit responsible for coordinating the data flow of a signature event. It allows for advanced filtering and signature overrides on the basis of the Event Risk Rating (ERR) feedback. ERR is used to control the level in which a user chooses to take actions in an effort to minimize false positives.

Signatures once stored in NVRAM, are now stored in the delta configuration file; thus, support for access control lists (ACLs) is no longer necessary.

Additional Risk Rating Algorithms

The ERR characterizes the risk of an attack and allows users to make decisions on the basis of the risk control signature event actions. To help further control signature event actions, the following additional rating categories are now supported:

- Attack Severity Rating (ASR)—Determines the severity of an attack. The attack-severity rating values are hard-coded in Cisco IOS IPS as follows: high, medium, low, and informational. The ASR can be changed through the **alert-rating** command. See the "Tuning Signature Parameters" section on page 14 for more information on changing the ASF.
- Signature Fidelity Rating (SFR)—Determines the confidence level of detecting a true positive. The SFR can be changed through the **fidelity-rating** command. See the "Tuning Signature Parameters" section on page 14 for more information on changing the SFR.

- Target Value Rating (TVR)—Allows users to develop security policies that can be more strict for some resources than others. The security policy is applied to a table of hosts that are protected by Cisco IOS IPS. A host can be a single IP address or a range of IP addresses with an associated target value rating. See the "Setting the Target Value Rating" section on page 21 for more information.

Signature Update Accessibility

To help detect the latest vulnerabilities, Cisco provides the following signature update options:

- Download the latest signatures from Cisco.com at the following URL:

  http://tools.cisco.com/support/downloads/go/Model.x?mdfid=281442967&mdfLevel=Software%20Family&treeName=Security&modelName=Cisco%20IOS%20Intrusion%20Prevention%20System%20Feature%20Software&treeMdfId=268438162. Ensure that you have your Cisco userid, and password to access this URL.
- Configure automatic signature updates through the **ip ips auto-update** command. Updates can be configured to run on the basis of a preset time. See the "Enabling Automatic Signature Updates" section on page 22" for more information.
- Issue the **copy** *url* **idconf** command to instruct the router where to load a signature file. (The file can be saved in a location specified through the **ip ips config location** command.)

Upgrade IPS Signatures Directly from Cisco.com

The Direct Download from CCO capability in IOS IPS feature allows administrator to configure the router to receive future periodic signature downloads automatically to eliminate the manual maintenance efforts and costs of changing or tuning IPS signatures whenever a new IPS signature update is posted. See the "Enabling Automatic Signature Updates" section on page 22 for more information.

The Direct Download from CCO capability in IOS IPS feature also allows an administrator to use the CLI to specify, download and upgrade to a new signatures posted for the IOS directly from Cisco.com. See the "Upgrading Signatures Directly from Cisco.com" section on page 24for more information.

Previously, the IPS signature file was downloaded and updated on the router either manually or through a management tool like Cisco Security Manager (CSM) or Security Device Manager (SDM), which both have access to the update area on Cisco.com.

Preserving Configured Signature Tunings on the Local Router

Most IPS devices and applications provide either a single default configuration or multiple default configurations. Using one of these default configurations is an ideal starting point for deploying IPS. When IOS IPS is deployed, parameters such as severity, active status or event actions of certain signatures need to be tuned to meet the requirements of an enterprise network traffic profile.

Once the **ip ips enable-clidelta** command is enabled, a local cli-delta.xmz file is generated containing the local tuning signatures configured through the CLI. The settings in the clidelta.xmz file take precedence when a globally administered delta signature update, contained in the iosips-sig-delta.xmz file, is sent from a central repository and applied to the configuration of the local router. See the "Tuning Signatures per Signature ID" section on page 14 for more information about the configuration of this feature.

How to Use Cisco IOS IPS 5.x Signature Format Support and Usability Enhancements

Retiring All Signatures and Selecting a Category of Signatures

Router memory and resource constraints prevent a router from loading all Cisco IOS IPS signatures. Thus, it is recommended that you load only a selected set of signatures that are defined by the categories. Because the categories are applied in a “top-down” order, you should first retire all signatures, followed by “unretiring” specific categories. Retiring signatures enables the router to load information for all signatures, but the router does not build the parallel scanning data structure.

Retired signatures are not scanned by Cisco IOS IPS, so they do not fire alarms. If a signature is irrelevant to your network or if you want to save router memory, you should retire signatures, as appropriate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips signature-category**
4. **category** *category* [*subcategory*]
5. **retired** {**true** | **false**}
6. **exit**
7. **category** *category* [*subcategory*]
8. **retired** {**true** | **false**}
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips signature-category** **Example:** `Router(config)# ip ips signature-category`	Enters enters IPS category configuration mode.
Step 4	**category** *category* [*subcategory*] **Example:** `Router(config-ips-category)# category all`	Specifies that all categories (and all signatures) are retired in the following step and enters IPS category action configuration mode.
Step 5	**retired** {**true** | **false**} **Example:** `Router(config-ips-category-action)# retired true`	Specifies that the router should retire all categories (and all signatures). • **true**—Retires all signatures within a given category. • **false** —"Unretires" all signatures within a given category.
Step 6	**exit** **Example:** `Router(config-ips-category-action)# exit`	Exits IPS category action configuration mode.
Step 7	**category** *category* [*subcategory*] **Example:** `Router(config-ips-category)# category ios_ips basic`	Specifies the basic category (and a set of signatures) that are to be "unretired" in the following step.
Step 8	**retired** {**true** | **false**} **Example:** `Router(config-ips-category-action)# retired false`	Specifies that all signatures within the basic category are to be unretired; that is, signatures are enabled for the basic category.
Step 9	**exit** **Example:** `Router(config-ips-category-action)# exit`	Exits IPS category action and IPS category configuration modes.

What to Do Next

After you have configured the basic category, you should enable IPS on your router. See the "Configuring Cisco IOS IPS on Your Router" section on page 9 for more information.

You can customize (or tune) either the entire category or individual signatures within a category to addresses the needs of your network. See the "Tuning Signature Parameters" section on page 14, for more information.

Configuring Cisco IOS IPS on Your Router

After you have set up a "load definition" for the signatures to be copied to the idconf, you must configure an IPS rule name. Use this task to configure an IPS rule name and start the IPS configuration.

You can also use this task to configure a Cisco IOS IPS signature location, which tells Cisco IOS IPS where to save signature information.

The configuration location is used to restore the IPS configuration in case the router reboots or IPS is disabled or reenabled. Files, such as signature definition, signature-type definitions, and signature category information, are written in XML format, compressed, and saved to the specified IPS signature location.

SUMMARY STEPS

1. **enable**
2. **mkdir flash:/ips5**
3. **configure terminal**
4. **ip ips name** *ips-name*
5. **ip ips config location** *url*
6. **interface** *type name*
7. **ip ips** *ips-name* {**in** | **out**}
8. **exit**
9. **show ip ips configuration**
10. **show ip ips signature** *count*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**mkdir flash:/ips5** **Example:** `Router# mkdir flash:/ips5`	Create a directory for which Cisco IOS IPS saves signature information. **Note** The directory location is specified through the **ip ips config location** command.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**ip ips name** *ips-name* **Example:** `Router(config)# ip ips name myips`	Creates an IPS rule.
Step 5	**ip ips config location** *url* **Example:** `Router(config)# ip ips config location flash:/ips5`	Specifies the location where Cisco IOS IPS saves the signature information, and, if necessary, access the signature configuration information. **Note** You must specify a location; otherwise, the signatures are not saved. **Note** If the specified location is a URL, such as an FTP server, the user must have writer privileges.
Step 6	**interface** *type name* **Example:** `Router(config)# interface gigbitEthernet 0/0`	Identifies the interface in which to enable Cisco IOS IPS and enters interface configuration mode.
Step 7	**ip ips** *ips-name* {**in** | **out**} **Example:** `Router(config-if)# ip ips MYIPS in`	Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines. **Note** Whenever signatures are replaced or merged, the router prompt is suspended while the signature engines for the newly added or merged signatures are being built. The router prompt is available again after the engines are built. Depending on your platform and how many signatures are being loaded, building the engine can take up to several minutes. It is recommended that you enable logging messages to monitor the engine building status.

	Command or Action	Purpose

Step 8	**exit**
	
	Example:
	`Router(config-if)# exit`
	`Router(config)# exit`
		Exits interface and global configuration modes.

Step 9	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	(Optional) Verifies that Cisco IOS IPS is properly configured.
Step 10	**show ip ips signature** *count* **Example:** `Router# show ip ips signature`	(Optional) Verifies the number of signatures that are loaded into each signature micro engine (SME).

Examples

The following sample output displays the number of signatures that have been loaded into each SME:

```
Router# show ip ips signature count

Cisco SDF release version S247.0
Trend SDF release version V1.2
Signature Micro-Engine: multi-string
Total Signatures: 7
Enabled: 7
Retired: 2
Compiled: 5
Signature Micro-Engine: service-http
Total Signatures: 541
Enabled: 284
Retired: 336
Compiled: 205
Signature Micro-Engine: string-tcp
Total Signatures: 487
Enabled: 332
Retired: 352
Compiled: 135
Signature Micro-Engine: string-udp
Total Signatures: 50
Enabled: 3
Retired: 23
Compiled: 27
Signature Micro-Engine: state
Total Signatures: 26
Enabled: 15
Retired: 23
Compiled: 3
Signature Micro-Engine: atomic-ip
Total Signatures: 140
Enabled: 87
Retired: 93
Compiled: 46
Inactive - invalid params: 1
Signature Micro-Engine: string-icmp
Total Signatures: 2
Enabled: 0
```

```
Retired: 1
Compiled: 1
Signature Micro-Engine: service-ftp
Total Signatures: 3
Enabled: 3
Compiled: 3
Signature Micro-Engine: service-rpc (INACTIVE)
Signature Micro-Engine: service-dns
Total Signatures: 1
Enabled: 1
Retired: 1
Signature Micro-Engine: normalizer
Total Signatures: 9
Enabled: 9
Compiled: 9
Total Signatures: 1266
Total Enabled Signatures: 741
Total Retired Signatures: 831
Total Compiled Signatures: 434
Total Signatures with invalid parameters: 1
```

Loading a Signature File into Cisco IOS IPS

Use this task to load signatures into Cisco IOS IPS. You may wish to load new signatures into Cisco IOS IPS if a signature (or signatures) with the current signatures are not providing your network with adequate protection from security threats.

Prerequisites

You must enable Cisco IOS IPS. See the "Configuring Cisco IOS IPS on Your Router" section on page 9" before loading new signatures.

Flexible Signatures: Ordered and Incremental

Each signature is complied incrementally into the scanning tables at the same time. Thus, Cisco IOS IPS can deactivate signatures that fail to compile. (Prior to Cisco IOS Release 12.4(11)T, Cisco IOS IPS deactivated the entire signature microengine (SME) if a single signature failed to compile.)

Signatures are loaded into the scanning table on the basis of importance. Parameters such as signature severity, signature fidelity rating, and time lapsed since signatures were last released allow Cisco IOS IPS to compile the most important signatures first, followed by less important signatures, thereby, creating a load order and prioritizing which signatures are loaded first.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips config location** *url*
4. **interface** *type name*
5. **ip ips** *ips-name* {**in** | **out**}
6. **exit**
7. **copy** *url* **idconf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips config location** *url* **Example:** `Router(config)# ip ips config location flash:/ips5`	Specifies the location where Cisco IOS IPS saves the signature information, and, if necessary, access the signature configuration information.
Step 4	**interface** *type name* **Example:** `Router(config)# interface gigbitEthernet 0/0`	Identifies the interface in which to enable Cisco IOS IPS.
Step 5	**ip ips** *ips-name* {**in** | **out**} **Example:** `Router(config-if)# ip ips MYIPS in`	Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines.

Step 6	**exit**
	
	Example:
	`Router(config-if)# exit`
	`Router(config)# exit`
		Exits interface and global configuration modes.

Step 7	**copy** *url* **idconf** **Example:** `Router# copy tftp://tftp_server/sig.xml idconf`	Loads signatures into Cisco IOS IPS. After the signatures are loaded, all signature information is saved to the location specified through the **ip ips config location** command.

Enabling IPS Regex Table Chaining

Regex Table Chaining is an enhancement to Cisco IOS IPS in Cisco IOS Release 15.0(1)M, that allows the IPS to chain multiple regular (regex) tables together when signatures are being loaded. This functionality is enabled by the **ip ips memory regex chaining** command. Enabling regex table chaining allows three regex tables to be chained. There is a slight performance degradation in IPS scanning time due to the scanning of multiple tables.

When a user tries to load a specific set of signatures that does not fit using a single table, compilation errors result.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips memory regex chaining**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips memory regex chaining** **Example:** `Router(config)# ip ips memory regex chaining`	Enables IPS regex table chaining when signatures are being loaded.

Tuning Signature Parameters

You can tune signature parameters on the basis of a signature ID (for an individual signature), or you can tune signature parameters on the basis of a category (that is, all signatures that are within a specified category). To tune signature parameters, use the following tasks, as appropriate:

- Tuning Signatures per Signature ID, page 14
- Tuning Signatures per Category, page 17

Note Some changes to the signature definitions are not shown in the run time config because the changes are recorded in the sigdef-delta.xml file, which can be located through the **ip ips config location** command.

Tuning Signatures per Signature ID

Use this task to change default signature parameters for a specified signature ID.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips enable-clidelta** (optional)
4. **ip ips signature-definition**
5. **signature** *signature-id* [*subsignature-id*]

6. **engine** (optional)
7. **event-action** *action*
8. **exit**
9. **alert-severity** {**high** | **medium** | **low** | **informational**} (optional)
10. **fidelity-rating** *rating* (optional)
11. **status** (optional)
12. **enabled** {**true** | **false**} (optional)
13. **exit**
14. **show ip ips signature** (optional)
15. **show ip ips sig-clidelta** (optional)

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips enable-clidelta** **Example:** `Router(config)# ip ips enable-clidelta`	(Optional) Enables the signature tuning settings in the clidelta.xmz file on the router to take precedence over the signature settings in the IPS iosips-sig-delta.xmz file.
Step 4	**ip ips signature-definition** **Example:** `Router(config)# ip ips signature-definition`	Enters signature-definition-signature configuration mode.
Step 5	**signature** *signature-id* [*subsignature-id*] **Example:** `Router(config-sigdef-sig)# signature 9000:0`	Specifies a signature for which the CLI user tunings are changed and enters signature-definition-action configuration mode.
Step 6	**engine** **Example:** `Router(config-sigdef-action)# engine`	(Optional) Enters signature-definition-action-engine configuration mode, which allows you to change router actions for a specified signature.

	Command or Action	Purpose
Step 7	**event-action** *action* **Example:** `Router(config-sigdef-action-engine)# event-action deny-attacker-inline`	Changes router actions for a specified signature. The *action* argument can be any of the following options: • **deny-attacker-inline** • **deny-connection-inline** • **deny-packet-inline** • **produce-alert** • **reset-tcp-connection** **Note** Signature event actions must be entered on a single line. **Note** You must enter the **engine** command before issuing this command.
Step 8	**exit** **Example:** `Router(config-sigdef-action-engine)# exit`	Exits the signature-definition-action-engine configuration mode. This step is required only if the **engine** and **event-action** commands are issued.
Step 9	**alert-severity** {**high** | **medium** | **low** | **informational**} **Example:** `Router(config-sigdef-action)# alert-severity medium`	(Optional) Changes the alert severity rating for a given signature.
Step 10	**fidelity-rating** *rating* **Example:** `Router(config-sigdef-action)# fidelity-rating`	(Optional) Changes the signature fidelity rating for a given signature.
Step 11	**status** **Example:** `Router(config-sidef-action)# status`	(Optional) Enters the signature-definition-status configuration mode, which allows you to change the enabled status of a signature.
Step 12	**enabled** {**true** | **false**} **Example:** `Router(config-sigdef-status)# enabled true`	(Optional) Changes the enabled status of a given signature or signature category.
Step 13	**exit** **Example:** `Router(config-sigdef-status)# exit`	Returns to EXEC mode.

	Command or Action	Purpose
Step 14	**show ip ips signature** **Example:** `Router# show ip ips signature`	(Optional) Verifies the signature changes that have been made.
Step 15	**show ip ips sig-clidelta** **Example:** `Router# show ip ips sig-clidelta`	(Optional) Displays the signature parameter tunings configured using the CLI, which are stored in the iosips-sig-clidelta.xmz signature file.

Tuning Signatures per Category

Use this task to change default signature parameters for a category of signatures. Categories such as operating systems; Layer 2, Layer 3, or Layer 4 protocols; or service-based categories can be configured to provide wider changes to a group of signatures.

Tip Category configuration information is processed in the order that it is entered. Thus, it is recommended that the process of retiring all signatures. See the "Retiring All Signatures and Selecting a Category of Signatures" section on page 7" before all other category tuning.

If a category is configured more than once, the parameters entered in the second configuration are added to or replace the previous configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips signature-category**
4. **category** *category* [*subcategory*]
5. **event-action** *action*
6. **alert-severity** {**high** | **medium** | **low** | **informational**}
7. **fidelity-rating** *rating*
8. **enabled** {**true** | **false**}
9. **retired** {**true** | **false**}
10. **exit**
11. **show ip ips signature**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips signature-category** **Example:** `Router(config)# ip ips signature-category`	Enters IPS category (config-ips-category) configuration mode.
Step 4	**category** *category* [*subcategory*] **Example:** `Router(config-ips-category)# category attack adware/spyware`	Specifies a category that is to be used for multiple signature actions or conditions and enters IPS category action configuration mode.
Step 5	**event-action** *action* **Example:** `Router(config-ips-category-action)# event-action produce-alert`	Changes router actions for a specified signature category. The *action* argument can be any of the following options: • **deny-attacker-inline** • **deny-connection-inline** • **deny-packet-inline** • **produce-alert** • **reset-tcp-connection** **Note** Event actions associated with a category can be entered separately or on a single line.
Step 6	**alert-severity** {**high** | **medium** | **low** | **informational**} **Example:** `Router(config-ips-category-action)# alert-severity medium`	(Optional) Changes the alert severity rating for a given signature category.
Step 7	**fidelity-rating** *rating* **Example:** `Router(config-ips-category-action)# fidelity-rating`	(Optional) Changes the signature fidelity rating for a signature given category.
Step 8	**enabled** {**true** | **false**} **Example:** `Router(config-ips-category-action)# enabled true`	(Optional) Changes the enabled status of a given signature or signature category.

	Command or Action	Purpose
Step 9	**retired** {**true** | **false**} **Example:** `Router(config-ips-category-action)# retired true`	(Optional) Specifies whether or not the router should retire a signature category.

Step 10	**exit**
	
	Example:
	`Router(config-ips-category-action)# exit`
	`Router(config-ips-category)# exit`
	`Router(config)# exit`
		Returns to EXEC mode, which allows you to later verify the configuration.

Step 11	**show ip ips signature** **Example:** `Router# show ip ips signature`	(Optional) Verifies the signature category changes that have been made.

Enabling Signature Tunings Inheritance

When new signatures are replacing older signatures, Cisco IOS IPS provides the **ip ips inherit-obsolete-tunings** command to enable new signatures to obsolete older signatures and inherit the event-action and enabled parameters of the obsolete tuning values, without the need to manually tune the new signatures. This functionality is called signature tuning inheritance. All other parameter changes including the “Retire” parameter is ignored.

After you enter the command, the screen displays a warning message asking you to clarify the intended usage and then asks whether you accept the configuration or not. By default, old signatures are not inherited by new signatures.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips inherit-obsolete-tunings**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips inherit-obsolete-tunings** **Example:** `Router(config)#ip ips inherit-obsolete-tunings`	Enables the inheritance of the tunings of the enabled and event-action parameters from obsolete signatures to new signatures in an IPS,

Setting an IPS Memory Threshold

When a router is powered up, 90 percent of the available memory is allocated to IPS-related activities. The remaining 10 percent is referred to as the IPS Memory Threshold—the amount of free memory unavailable to the IPS. Cisco IOS IPS allows the IPS memory threshold to be set to a different value using the **ip ips memory threshold** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips memory threshold** *megabytes*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips memory threshold** *megabytes* **Example:** `Router(config)# ip ips memory threshold 50`	Specifies an IPS memory threshold, or the amount of free memory unavailable to the IPS. • The units are in megabytes (MB). • This example specifies that Cisco IOS IPS cannot consume more memory if the remaining (free) memory becomes less than 50 MB.

Setting the Target Value Rating

Use this task to set the target value rating, which allows users to develop security policies that can be more strict for some resources than others. The security policy is applied to a table of hosts that are protected by Cisco IOS IPS. A host can be a single IP address or a range of IP addresses with an associated target value rating.

Note Changes to the target value rating is not shown in the run time config because the changes are recorded in the seap-delta.xml file, which can be located through the **ip ips config location** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips event-action-rules**
4. **target-value** {**mission-critical** | **high** | **medium** | **low**} **target-address** *ip-address* [**/***nn* | **to** *ip-address*]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips event-action-rules** **Example:** `Router(config)# ip ips event-action-rules`	Enters the config-rule configuration mode, which allows users to change the target value rating.
Step 4	**target-value** {**mission-critical** | **high** | **medium** | **low**} **target-address** *ip-address* [**/***nn* | **to** *ip-address*] **Example:** `Router(config-rul)# target-value medium target-address 10.12.100.53`	Sets the target value rating for a host.
Step 5	**exit** **Example:** `Router(config-rul)# exit`	Exits config-rule configuration mode.

Enabling Automatic Signature Updates

Automatic signature updates allow users to override the existing configuration and automatically keep signatures up to date on the basis of a preset time, which can be configured to a preferred setting.

Time can be updated through the hardware clock or the configurable software clock (which ever option is available on your system). Although Network Time Protocol (NTP) is typically used for automated time synchronization, Cisco IOS IPS updates use the local clock resources as a reference for update intervals. Thus, NTP should be configured to update the local time server of the router, as appropriate.

Perform this task to enable Cisco IOS IPS to automatically update the signature file on the router from the local server or receive future periodic signature downloads automatically from Cisco.com.

Automatic Signature Update Guidelines

When enabling automatic signature updates, it is recommended that you ensure the following configuration guidelines have been met:

- The router's clock is set up with the proper relative time.
- The frequency for Cisco IOS IPS to obtain updated signature information has been defined.
- If updates are being retrieved from a local server, obtain the URL from which the Cisco IOS IPS signature configuration files are retrieved.
- Optionally, the username and password for which to access the files from the server have been specified.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips auto-update**
4. **cisco**
5. **occur-at** { [**monthly** | **weekly**] *days minutes hours*}
6. **username** *name* **password** *password*
7. **url** *url*
8. **exit**
9. **show ip ips auto-update**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip ips auto-update** **Example:** `Router(config)# ip ips auto-update`	Enables automatic signature updates for Cisco IOS IPS and enters IPS auto-update configuration mode.
Step 4	**cisco** **Example:** `Router(config-ips-auto-update)# cisco`	(Optional) Enables automatic signature updates from Cisco.com. **Note** This command cannot be used together with the **url** command in Step 7.
Step 5	**occur-at** {[**monthly** | **weekly**] *days minutes hours*} **Example:** `Router(config-ips-auto-update)# occur-at weekly 4 23 23`	(Optional) Defines a preset time for which the Cisco IOS IPS automatically obtains updated signature information.
Step 6	**username** *name* **password** *password* **Example:** `Router(config-ips-auto-update)# username myips password secret`	(Optional) Defines a username and password for the automatic signature update function.
Step 7	**url** *url* **Example:** `Router(config-ips-auto-update)# url tftp://192.168.0.2/username1/ips-auto-update/IOS_reqSeq-dw.xml`	(Optional) URL from the local server on which the router retrieves the Cisco IOS IPS signature configuration files. **Note** This command cannot be used together with the **cisco** command in Step 4.

Step 8	**exit**
	
	Example:
	`Router(config-ips-auto-update)# exit`
	`Router(config)# exit`
		Exits IPS auto-update and global configuration modes.

Step 9	**show ip ips auto-update** **Example:** `Router# show ip ips auto-update`	Verifies the automatic signature update configuration.

Upgrading Signatures Directly from Cisco.com

Perform this task to specify, download and upgrade to a new IPS signature file posted for the IOS directly from Cisco.com.

This task eliminates the need for an administrator to download the latest signature file from CCO manually to a local HTTP, FTP, or TFTP server and next download this signature file to the router (from that local HTTP, FTP, or TFTP server).

SUMMARY STEPS

1. **enable**
2. **ips signature update cisco**
3. **configure terminal**
4. **ida-client server url** *url*
5. **exit**
6. **show ip ips configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ips signature update cisco** {**next** | **latest** | *signature*} [**username** *name* **password** *password*] **Example:** `Router# ip ips update cisco IOS_reqSeq-dw.xml`	Initiates a one-time download of an IPS signatures from Cisco.com. The **next** keyword specifies the next signature file version from the current signature file on the router. The **latest** keyword specifies the IOS IPS to search for the latest signature file. The *signature* argument specifies a specific signature file on Cisco.com. (Optional) The username keyword and name argument and password keyword and password argument is for the automatic signature update function. **Note** If the username and password is not specified, then the username and password that was specified in the "Enabling Automatic Signature Updates" section on page 22 is used. If the username and password are not configured in this section, a user name and password must be configured for updating signatures directly from Cisco.com.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 4	**ida-client server url** *url* **Example:** `Router(config)# ida-client server url https://www.cisco.com/cgi-bin/front.x/ida/locator/locator.pl`	Specifies the IDA-server URL that the IDA client communicates with to download files from the Cisco.com server. Enter the following URL: https://www.cisco.com/cgi-bin/front.x/ida/locator/locator.pl
Step 5	**password encryption aes** **Example:** `Router(config)# password encryption aes`	Enables the encryption of the password with a type 6 encrypted preshared key.
Step 6	**key config-key password-encryption** **Example:** `Router(config)# key config-key password-encryption`	Configures the encrypted preshared key that is used to encrypt all other keys in the router.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 8	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	Verifies that Cisco IOS IPS is properly configured.

Monitoring Cisco IOS IPS Signatures through Syslog Messages or SDEE

Cisco IOS IPS provides two methods to report IPS intrusion alerts—Cisco IOS logging (syslog) and SDEE. Perform this task to enable SDEE to report IPS intrusion alerts. See the "Troubleshooting and Fault Management" feature module for more information on configuring syslog messages.

SDEE Overview

SDEE is an application-level communication protocol that is used to exchange IPS messages between IPS clients and IPS servers. SDEE is always running, but it does not receive and process events from IPS unless SDEE notification is enabled. If SDEE notification is not enabled and a client sends a request, SDEE responds with a fault response message, indicating that notification is not enabled.

Storing SDEE Events in the Buffer

When SDEE notification is enabled (through the **ip ips notify sdee** command), 200 events can automatically be stored in the buffer. When SDEE notification is disabled, all stored events are lost. A new buffer is allocated when the notifications are reenabled.

When specifying the size of an events buffer, note the following functionality:

- It is circular. When the end of the buffer is reached, the buffer starts overwriting the earliest stored events. (If overwritten events have not yet been reported, a buffer overflow notice is received.)
- If a new, smaller buffer is requested, all events that are stored in the previous buffer is lost.

- If a new, larger buffer is requested, all existing events are saved.

Prerequisites

To use SDEE, the HTTP server must be enabled (through the **ip http server** command). If the HTTP server is not enabled, the router cannot respond to the SDEE clients because it cannot not "see" the requests.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips notify sdee**
4. **ip sdee events** *events*
5. **ip sdee subscriptions** *subscriptions*
6. **ip sdee messages** *messages*
7. **ip sdee alerts** *alerts*
8. **exit**
9. **show ip sdee** {[**alerts**] [**all**] [**errors**] [**events**] [**configuration**] [**status**] [**subscriptions**]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips notify sdee** **Example:** `Router(config)# ip ips notify sdee`	Enables SDEE event notification on a router.
Step 4	**ip sdee events** *events* **Example:** `Router(config)# ip sdee events 500`	(Optional) Sets the maximum number of SDEE events that can be stored in the event buffer. • Maximum value: 1000 events. **Note** By default, 200 events can be stored in the buffer when SDEE is enabled. When SDEE is disabled, all stored events are lost; a new buffer is allocated when the notifications are reenabled.

	Command or Action	Purpose
Step 5	**ip sdee subscriptions** *subscriptions* **Example:** `Router(config)# ip sdee subscriptions 1`	(Optional) Sets the maximum number of SDEE subscriptions that can be open simultaneously. • Valid value ranges from 1 to 3.
Step 6	**ip sdee messages** *messages* **Example:** `Router(config)# ip sdee messages 500`	(Optional) Sets the maximum number of SDEE messages that can be stored in the buffer at one time.
Step 7	**ip sdee alerts** *alerts* **Example:** `Router(config)# ip sdee alerts 2000`	(Optional) Sets the maximum number of SDEE alerts that can be stored in the buffer at one time.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 9	**show ip sdee** {[**alerts**] [**all**] [**errors**] [**events**] [**configuration**] [**status**] [**subscriptions**]} **Example:** `Router# show ip sdee configuration`	(Optional) Verifies SDEE configuration information and notification functionality.

Troubleshooting Tips

To print out new SDEE alerts on the router console, issue the **debug ip sdee** command.

To clear the event buffer or SDEE subscriptions from the router (which helps with error recovery), issue the **clear ip sdee** command.

Configuration Examples for Cisco IOS IPS 5.x Signature Format Support and Usability Enhancements

Cisco IOS IPS Configuration: Example

The following example shows how to enable and verify Cisco IOS IPS on your router:

```
Router# mkdir flash:/ips5
Create directory filename [ips5]?
Created dir flash:/ips5
Router#
Router#
```

```
Router#
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip ips name MYIPS
Router(config)# ip ips config location flash:/ips5
Router(config)# ip ips signature-category
Router(config-ips-category)# category all
Router(config-ips-category-action)# retired true
Router(config-ips-category-action)# exit
Router(config-ips-category)# category ios_ips advanced
Router(config-ips-category-action)# retired false
Router(config-ips-category-action)# exit
Router(config-ips-category)# exit
Do you want to accept these changes? [confirm]
Router(config)# d
*Nov 14 2006 17:16:42 MST: Applying Category configuration to signatures ..
Router(config)#
Router(config)# do show ip interface brief
Interface                IP-Address      OK?    Method  Status                  Protocol
GigabitEthernet0/0       10.0.20.120     YES    NVRAM   up                      up
GigabitEthernet0/1       10.12.100.120   YES    NVRAM   administratively down  down
NVI0                     unassigned      NO     unset   up                      up
Router(config)#
Router(config)# interface gigabits 0/0
Router(config-if)# ip ips MYIPS in
Router(config-if)#
*Nov 14 2006 17:17:07 MST: %IPS-6-ENGINE_BUILDS_STARTED: 17:17:07 MST Nov 14 2006
*Nov 14 2006 17:17:07 MST: %IPS-6-ENGINE_BUILDING: atomic-ip - 3 signatures - 1 of 13
engines
*Nov 14 2006 17:17:07 MST: %IPS-6-ENGINE_READY: atomic-ip - build time 0 ms - packets for
this engine will be scanned
*Nov 14 2006 17:17:07 MST: %IPS-6-ALL_ENGINE_BUILDS_COMPLETE: elapsed time 0 ms
Router(config-if)#
Router(config-if)# ip ips MYIPS out
Router(config-if)#
Router(config-if)#
Router(config-if)#^Z
Router#
*Nov 14 2006 17:17:23 MST: %SYS-5-CONFIG_I: Configured from console by cisco on console
Router# wr
Building configuration...
[OK]
Router#
Router# show ip ips signature count
Cisco SDF release version S0.0

Signature Micro-Engine: multi-string (INACTIVE)
Signature Micro-Engine: service-http (INACTIVE)
Signature Micro-Engine: string-tcp (INACTIVE)
Signature Micro-Engine: string-udp (INACTIVE)
Signature Micro-Engine: state (INACTIVE)
Signature Micro-Engine: atomic-ip
     Total Signatures: 3
       Enabled: 0
       Compiled: 3
Signature Micro-Engine: string-icmp (INACTIVE)
Signature Micro-Engine: service-ftp (INACTIVE)
Signature Micro-Engine: service-rpc (INACTIVE)
Signature Micro-Engine: service-dns (INACTIVE)
Signature Micro-Engine: normalizer (INACTIVE)
Signature Micro-Engine: service-smb-advanced (INACTIVE)
Signature Micro-Engine: service-msrpc (INACTIVE)
     Total Signatures: 3
     Total Enabled Signatures: 0
```

```
     Total Retired Signatures: 0
     Total Compiled Signatures: 3
Router#
Router# copy flash:IOS-S258-CLI-kd.pkg idconf
*Nov 14 2006 17:19:47 MST: %IPS-6-ENGINE_BUILDS_STARTED: 17:19:47 MST Nov 14 2006
*Nov 14 2006 17:19:47 MST: %IPS-6-ENGINE_BUILDING: multi-string - 3 signatures - 1 of 13
engines
*Nov 14 2006 17:19:47 MST: %IPS-6-ENGINE_READY: multi-string - build time 4 ms - packets
for this engine will be scanned
*Nov 14 2006 17:19:47 MST: %IPS-6-ENGINE_BUILDING: service-http - 611 signatures - 2 of 13
engines
*Nov 14 2006 17:20:00 MST: %IPS-6-ENGINE_READY: service-http - build time 12932 ms -
packets for this engine will be scanned
*Nov 14 2006 17:20:00 MST: %IPS-6-ENGINE_BUILDING: string-tcp - 864 signatures - 3 of 13
engines
*Nov 14 2006 17:20:02 MST: %IPS-6-ENGINE_READY: string-tcp - build time 2692 ms - packets
for this engine will be scanned
*Nov 14 2006 17:20:02 MST: %IPS-6-ENGINE_BUILDING: string-udp - 74 signatures - 4 of 13
engines
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: string-udp - build time 316 ms - packets
for this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_BUILDING: state - 28 signatures - 5 of 13 engines
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: state - build time 24 ms - packets for
this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_BUILDING: atomic-ip - 252 signatures - 6 of 13
engines
*Nov 14 2006 17:20:03 MST: %IPS-4-META_ENGINE_UNSUPPORTED: atomic-ip 2154:0 - this
signature is a component of the unsupported META engine
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: atomic-ip - build time 232 ms - packets
for this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_BUILDING: string-icmp - 3 signatures - 7 of 13 e
Router# engines
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: string-icmp - build time 12 ms - packets
for this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_BUILDING: service-ftp - 3 signatures - 8 of 13
engines
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: service-ftp - build time 8 ms - packets
for this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_BUILDING: service-rpc - 75 signatures - 9 of 13
engines
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: service-rpc - build time 80 ms - packets
for this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_BUILDING: service-dns - 38 signatures - 10 of 13
engines
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: service-dns - build time 20 ms - packets
for this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_BUILDING: normalizer - 9 signatures - 11 of 13
engines
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: normalizer - build time 0 ms - packets for
this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_BUILDING: service-msrpc - 22 signatures - 12 of
13 engines
*Nov 14 2006 17:20:03 MST: %IPS-6-ENGINE_READY: service-msrpc - build time 8 ms - packets
for this engine will be scanned
*Nov 14 2006 17:20:03 MST: %IPS-6-ALL_ENGINE_BUILDS_COMPLETE: elapsed time 16344 ms
Router#
Router#
Router# show ip ips signature count
Cisco SDF release version S258.0

Signature Micro-Engine: multi-string
     Total Signatures: 3
       Enabled: 3
       Retired: 3
```

```
Signature Micro-Engine: service-http
     Total Signatures: 611
       Enabled: 159
       Retired: 428
       Compiled: 183
Signature Micro-Engine: string-tcp
     Total Signatures: 864
       Enabled: 414
       Retired: 753
       Compiled: 111
Signature Micro-Engine: string-udp
     Total Signatures: 74
       Enabled: 1
       Retired: 44
       Compiled: 30
Signature Micro-Engine: state
     Total Signatures: 28
       Enabled: 16
       Retired: 25
       Compiled: 3
Signature Micro-Engine: atomic-ip
     Total Signatures: 252
       Enabled: 56
       Retired: 148
       Compiled: 103
       Inactive - invalid params: 1
Signature Micro-Engine: string-icmp
     Total Signatures: 3
       Enabled: 0
       Retired: 2
       Compiled: 1
Signature Micro-Engine: service-ftp
     Total Signatures: 3
       Enabled: 1
       Compiled: 3
Signature Micro-Engine: service-rpc
            Total Signatures: 75
       Enabled: 44
       Retired: 44
       Compiled: 31
Signature Micro-Engine: service-dns
     Total Signatures: 38
       Enabled: 30
       Retired: 5
       Compiled: 33
Signature Micro-Engine: normalizer
     Total Signatures: 9
       Enabled: 8
       Retired: 5
       Compiled: 4
Signature Micro-Engine: service-smb-advanced (INACTIVE)
Signature Micro-Engine: service-msrpc
     Total Signatures: 22
       Enabled: 22
       Retired: 22
```

Enabling Automatic Signature Updates: Example

The following example shows how to configure automatic signature updates and issue the **show ip ips auto-update** command to verify the configuration. In this example, the signatures are pulled from the TFTP server at the start of every hour or every day, Sunday through Thursday. (Note that adjustments are made for months without 31 days and daylight savings time.)

```
Router# clock set ?
    hh:mm:ss Current Time
Router# clock set 10:38:00 20 apr 2006
Router#
*Apr 20 17:38:00.000: %SYS-6-CLOCKUPDATE: System clock has been updated from 10:37:55 MST
Thu Apr 20 2006 to 10:38:00 MST Thu Apr 20 2006, configured from console by cisco on
console.

Router(config)# ip ips auto-update
Router(config-ips-auto-update)# occur-at 0 0-23 1-31 1-5
Router(config-ips-auto-update)# $s-auto-update/IOS_reqSeq-dw.xml
Router(config-ips-auto-update)#^Z
Router#
*May 4 2006 15:50:28 MST: IPS Auto Update: setting update timer for next update: 0 hrs 10
min
*May 4 2006 15:50:28 MST: %SYS-5-CONFIG_I: Configured from console by cisco on console
Router#
Router# show ip ips auto-update

IPS Auto Update Configuration
    URL : tftp://192.168.0.2/jdoe/ips-auto-update/IOS_reqSeq-dw.xml
    Username : not configured
    Password : not configured
    Auto Update Intervals
      minutes (0-59) : 0
      hours (0-23) : 0-23
      days of month (1-31) : 1-31
      days of week: (0-6) : 1-5
```

Configuring and Verifying SDEE on your Router: Example

The following example shows how to configure and verify SDEE on your router:

```
Router(config)# ip ips notify SDEE
Router(config)# ip sdee event 500
Router(config)# ip sdee subscriptions 1
Router(config)# ip sdee messages 500
Router(config)# ip sdee alerts 2000
router(config)# exit
*Nov 9 21:41:33.171: %SYS-5-CONFIG_I: Configured from console by cisco on console
Router# show ip sdee all
Configured concurrent subscriptions: 1
No currently open subscriptions.
Alert storage: 2000 alerts using 560000 bytes of memory
Message storage: 500 messages using 212000 bytes of memory
SDEE Events
Time Type Description
Router#
```

Additional References

Related Documents

Related Topic	Document Title
Security commands	*Cisco IOS Security Command Reference*
Loading images and file systems	Loading and Managing System Images feature module.
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*

Standards

Standard	Title
No new or modified standards are supported by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco IOS IPS 5.x Signature Format Support and Usability Enhancements

Table 1 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 1 Feature Information for Cisco IOS 5.x Signature Format Support and Usability Enhancements

Feature Name	Releases	Feature Information
Cisco IOS IPS 5.x Signature Format and Usability Enhancements	12.4(11)T	This feature introduces support for Cisco IOS Intrusion Prevention System (IPS) version 5.0, which is a version-based signature definition XML format. Cisco IOS IPS 4.x format signatures are replaced by the 5.x format signatures that are used by all other Cisco IPS devices. The following commands were introduced or modified by this feature: **alert-severity**, **category**, **copy idconf enabled (IPS)**, **engine (IPS)**, **event-action**, **fidelity-rating**, **ip ips auto-update**, **ip ips config location**, **ip ips event-action-rules**, **ip ips signature-category**, **ip ips signature-definition**, **occur-at (ips-auto-update)**, **retired (IPS)**, **show ip ips auto-update**, **signature**, **status**, **target-value**, **url (ips-auto-update)**, **username (ips-autoupdate)**.
Cisco IOS IPS with Lightweight Signatures	15.0(1)M	This feature extends support for lightweight signatures in Cisco IOS IPS. Lightweight signatures allow the loading of a larger number of signatures simultaneously, without consuming significant additional memory or reducing the memory consumed by an existing signature set. The following sections provides information about this feature: • Cisco IOS IPS Signature Scanning with Lightweight Signatures, page 5 • Enabling IPS Regex Table Chaining, page 13 • Enabling Signature Tunings Inheritance, page 19 • Setting an IPS Memory Threshold, page 20 The following commands were introduced or modified by this feature: **ip ips inherit-obsolete tunings**, **ip ips memory regex chaining**, **ip ips memory threshold**.

Table 1 Feature Information for Cisco IOS 5.x Signature Format Support and Usability Enhancements

Feature Name	Releases	Feature Information
Direct Download from CCO capability in IOS IPS	15.1(1)T	This feature was introduced to allow an administrator to use the CLI to specify, download and upgrade new signatures posted for the IOS directly from Cisco.com. An administrator can also configure the router through the CLI to receive future periodic signature downloads automatically to eliminate the manual maintenance efforts and costs of changing or tuning IPS signatures whenever a new IPS signature update is posted. The following commands were introduced or modified by this feature: **cisco**, **ida-client server url**, **ip ips auto-update**, **ip signature update cisco**, **occur-at**.
Capability to save local delta changes on IOS routers	15.1(2)T	This feature was introduced to generate a local cli-delta.xmz file on the router containing the signature tuning settings configured through the CLI. This local file takes precedence when a globally administered delta signature update, contained in the IPS iosips-sig-delta.xmz file, is sent from a central repository and applied to the configuration of the local router. The following sections provide information about this feature: • Preserving Configured Signature Tunings on the Local Router, page 6 • Tuning Signatures per Signature ID, page 14 The following commands were introduced or modified: **ip ips enable-clidelta**, **show ip ips sig-clidelta**.