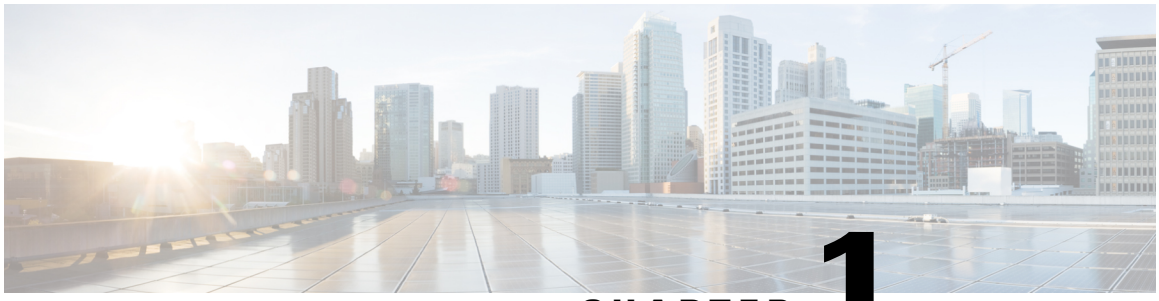
Cisco IOS Intrusion Prevention System Configuration Guide, Cisco IOS Release 15MT

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Configuring Cisco IOS Intrusion Prevention System

The Cisco IOS Intrusion Prevention System (IPS) uses the following methods to protect a network from internal and external attacks and threats:

- IPS signatures are dynamically updated and posted to Cisco.com on a regular basis so that customers can access signatures that help protect their network from the latest known network attacks.
- A parallel signature scanning engine is used to scan for multiple patterns within a signature microengine (SME) at any given time. IPS signatures are no longer scanned on a serial basis.
- Cisco IOS IPS supports both named and numbered extended access control lists (ACLs).

Note Cisco IOS IPS restructures and replaces the existing Cisco IOS Intrusion Detection System (IDS).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 table.

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

Prerequisites for Configuring Cisco IOS IPS

It is recommended that a new Cisco IOS image be loaded on your router before installing Cisco IOS IPS.

Compatibility with VMS IDS MC 2.3 and Cisco Router SDM

VPN/Security Management System (VMS) IDS Management Console (IDS MC) provides a web-based interface for configuring, managing, and monitoring multiple IDS sensors. Security Device Manager (SDM) is a web-based device-management tool that allows users to import and edit SDFs from Cisco.com to the router. VMS IDS MC is for network-wide management while SDM is for single-device management. It is strongly recommended that customers download the SDF to an IDS MC 2.3 network management device or an SDM.

Customers can choose to download the SDF to a device other than VMS IDS MC or SDM (such as a router) through command-line interface (CLI); however, this approach is not recommended because it requires that the customer know which signatures come from which signature engines.

Restrictions for Configuring Cisco IOS IPS

- Cisco IOS Intrusion Prevention System (IPS) does not support virtual templates.
- There is no separate license requirement for the IOS IPS Update feature. However; the output of the **show license feature** command displays the IOS IPS Update feature license as not enabled. This is a known issue.

Signature Support Deprecation

Effective Cisco IOS Release 12.3(8)T, the following signatures are no longer supported by Cisco IOS IPS:

- 1100 IP Fragment Attack (Attack, Atomic)—Triggers when any IP datagram is received with the “more fragments” flag set to 1 or if there is an offset indicated in the offset field. (To scan for application layer signatures across fragments, you can enable virtual fragment reassembly.)
- 1105 Broadcast Source Address (Compound/Attack)—Triggers when an IP packet with a source address of 255.255.255.255 is detected. This signature may be an indicator of an IP spoof attack or an attempt to subvert a firewall, proxy, or gateway.
- 1106 Multicast IP Source Address (Compound/Attack)—Triggers when an IP packet with a source address of 224.x.x.x is detected. This signature may be an indicator of an IP spoof attack or an attempt to subvert a firewall, proxy, or gateway.
- 8000 FTP Retrieve Password File (Attack, Atomic) SubSig If: 2101—Triggers on string “passwd” issued during an FTP session. May indicate that someone is attempting to retrieve the password file from a machine to try and gain unauthorized access to system resources.

Memory Impact on Low-End to Midrange Routers

Intrusion detection configuration on certain routers may not support the complete list of signatures because of lack of sufficient memory. Thus, the network administrator may have to select a smaller subset of signatures or choose to use the standard 100 (built-in) signatures with which the routers are shipped.

Action Configuration Through CLI No Longer Supported

Cisco IOS IPS actions (such as resetting the TCP connection) can no longer be configured through CLI. If you are using the attack-drop.sdf signature file, the signatures are preset with actions to mitigate the attack by dropping the packet and resetting the connection, if applicable. If you are using VMS or SDM to deploy signatures to the router, you must first tune the signatures to use the desired actions.

Any CLI that is issued to configure IPS actions is silently ignored.

Restrictions for Transparent Cisco IOS IPS

- Mixed-media bridging configurations are not supported. Only Ethernet media are supported.
- Layer 2 signatures are not supported.
- Multicast traffic is not processed.
- If more than two interfaces are assigned to a bridge group, any routers that are acting as first-hop gateways to hosts that are in the bridged network (the bridge group) must allow ICMP time-to-live (TTL) exceeded messages to pass.
- Spanning Tree Bridge Protocol Data Units (BPDU) and packets that are to be routed out of the bridge, if IRB is configured, are not inspected.

Information About Cisco IOS IPS

Cisco IOS IPS Overview

Cisco IOS IPS acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the device and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS 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. 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 can 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.

Security Device Event Exchange

Security Device Event Exchange (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 is saved.

Out-of-Order Packet Processing

Out-of-Order (OoO) packet processing support for Common Classification Engine (CCE) firewall application and CCE adoptions of the Intrusion Prevention System (IPS) allows packets that arrive out of order to be copied and reassembled in the correct order. The OoO packet processing reduces the need to retransmit dropped packets and reduces the bandwidth needed for the transmission of traffic on a network.

Transparent Cisco IOS IPS Overview

If customers want to protect their network through a typical Cisco IOS IPS device, they must manually readdress each of the statically defined devices on the trusted network. A transparent Cisco IOS IPS device allows customers to “drop” a Layer 3 IPS device in front of the devices that need to be protected. Thus, the tedious and costly overhead that is required to renumber devices on the trusted network is eliminated.

A transparent Cisco IOS IPS device acts as a Layer 3 IPS between bridged interfaces. (The current implementation of transparent IPS does not support Layer 2 IPS functionality; thus, IPS can act only as a Layer 3 device.)

Transparent Bridging Overview

If configured for bridging, a Cisco IOS device can bridge any number of interfaces. The device can complete basic bridging tasks such as learning MAC addresses on ports to restrict collision domains and running Spanning Tree Protocol (STP) to prevent looping in the topology.

Within bridging, a user can configure Integrated Routed Bridging (IRB), which allows a device to bridge on some interfaces while a Layer 3 Bridged Virtual Interface (BVI) is presented for routing. The bridge can determine whether the packet is to be bridged or routed on the basis of the destination of the Layer 2 or Layer 3 IP address in the packet. Configured with an IP address, the BVI can manage the device even if there is no interface configured for routing.

Transparent and Non-Transparent IPS Devices Configured on the Same Device

A transparent IPS device supports a bridge group virtual interface (BVI) for routing, so a packet that comes in on a bridged interface can be bridged or routed out of the BVI. This functionality allows a transparent IPS device and a non-transparent IPS device to be configured on the same device. The transparent IPS device operates on the bridged packets while the "normal" IPS device operates on the routed packets. For example, if you have six interfaces on your device and two of them are in a bridge group, you can simultaneously configure and run normal IPS inspection on the remaining four interfaces.

Users can also configure transparent IPS and a transparent firewall on the same device. For more information on the transparent firewall, see the document *Transparent Cisco IOS Firewall*.

Signature Definition File

Cisco IOS IPS allows customers to choose between any of the following options when loading IPS signatures onto a device:

- Loading the default, built-in signatures.
- Download the signature definition file (SDF) on the device by using the Cisco Router and Security Device Manager (SDM) to have the latest available detection of security threats. Go to the following link to download the SDF:

http://www.cisco.com/cgi-bin/tablebuild.pl/ios-sigup

Note SDM automatically recommends the SDF that should be used the first time when IPS is enabled on the device.

After the SDF is downloaded on to their device, the SDM can immediately begin scanning for new signatures.

An SDF has definitions for each signature that it contains. After signatures are loaded and complied onto a device running Cisco IOS IPS, IPS can begin detecting the new signatures immediately. If the default, built-in signatures that are shipped with the devices are not used, then one of three different types of SDFs can be selected for download, which are preconfigured for devices with memory requirements:

- **attack-drop.sdf** file (which is a static file that has 83 signatures) is used for devices with less than 128 MB of memory.
- **128MB.sdf** file (which has about 300 signatures) is used for devices with 128 MB or more memory.
- **256MB.sdf** file (which has about 500 signatures) is used for devices with 256 MB or more memory.

The **attack-drop.sdf**, **128MB.sdf**, and **256MB.sdf** files are available in flash on all Cisco access devices. One of these files can then be loaded directly from flash into the Cisco IOS IPS system. If flash is erased, the SDF file may also be erased. If a Cisco IOS image is copied to flash and there is a prompt to erase the contents

of flash before copying the new image, you might risk erasing the SDF file. If the SDF file is erased, the device refers to the built-in signatures within the Cisco IOS image.

The SDF file can also be downloaded onto your device from Cisco.com through SDM. SDF files can be loaded through IDS MC 2.3, which can be launched from CSM 3.0.

Note SDF files can be used only with 12.4(9)Tx or earlier Cisco IOS images and mainline images.

To help detect the latest vulnerabilities, Cisco provides signature updates on Cisco.com on a regular basis. Users can use VMS or SDM to download these signature updates, tune the signature parameters as necessary, and deploy the new SDF to a Cisco IOS IPS device.

Signature Microengines Overview and Lists of Supported Engines

Cisco IOS IPS uses SMEs to load the SDF and scan signatures. Signatures within the SDF are handled by a variety of SMEs. The SDF typically contains signature definitions for multiple engines. The SME typically corresponds to the protocol in which the signature occurs and looks for malicious activity in that protocol.

A packet is processed by several SMEs. Each SME scans for various conditions that can lead to a signature pattern match. When an SME scans the packets, it extracts certain values, searching for patterns within the packet through the regular expression engine. See "Lists of Supported Signature Engines" for a list of supported signature engines.

Lists of Supported Signature Engines

Note If the SDF contains a signature that requires an engine that is not supported, the engine is ignored and an error message is displayed. If a signature within a supported engine contains a parameter that is not supported, the parameter is ignored and an error message is displayed.

Table 1: Supported Signature Engines for Cisco IOS IPS

Signature Engine	Initial Supported Cisco IOS Release	Parameter Exceptions[1]
ATOMIC.L3.IP	12.3(8)T	—
ATOMIC.ICMP	12.3(8)T	—
ATOMIC.IPOPTIONS	12.3(8)T	—
ATOMIC.TCP	12.3(8)T	—
ATOMIC.UDP	12.3(8)T	—
SERVICE.DNS	12.3(8)T	—

Signature Engine	Initial Supported Cisco IOS Release	Parameter Exceptions[1]
SERVICE.HTTP	12.3(8)T	ServicePorts (applicable only in Cisco IOS Release 12.3(8)T)
SERVICE.FTP	12.3(8)T	ServicePorts
SERVICE.SMTP	12.3(8)T	ServicePorts
SERVICE.RPC	12.3(8)T	ServicePorts, Unique, and isSweep
STRING.ICMP	12.3(14)T	—
STRING.TCP	12.3(14)T	—
STRING.UDP	12.3(14)T	—

1 The following parameters, which are defined in all signature engines, are currently not supported: AlarmThrottle=Summarize (all other values are supported), MaxInspectLength, MaxTTL, Protocol, ResetAfterIdle, StorageKey, and SummaryKey.

The table below lists support for 100 signatures that are available in Cisco IOS IDS prior to Cisco IOS Release 12.3(8)T. As of Cisco IOS Release 12.3(8)T, these 100 signatures are a part of the Cisco IOS IPS built-in SDF. By default, signatures are loaded from this built-in SDF. The table above lists support for these 100 signatures under Cisco IOS IPS.

Note Because Cisco IOS IPS counts signatures on the basis of signature-id and subsignature-id, the 100 signatures under Cisco IOS IDS are counted as 132 signatures under Cisco IOS IPS.

Table 2: Support for Signatures Available in Cisco IOS IDS (Prior to 12.3(8)T)

Signature ID	Count	Signature Engine
1000-1006	7	ATOMIC.IPOPTIONS
1101, 1102	2	ATOMIC.L3.IP
1004, 1007	2	ATOMIC.L3.IP
2000-2012, 2150	14	ATOMIC.ICMP
2151, 2154	2	ATOMIC.L3.IP
3038-3043	6	ATOMIC.TCP
3100-3107	8	SERVICE.SMTP
3153, 3154	2	SERVICE.FTP

Signature ID	Count	Signature Engine
4050-4052, 4600	4	ATOMIC.UDP
6100-6103	4	SERVICE.RPC
6150-6155	6	SERVICE.RPC
6175, 6180, 6190	3	SERVICE.RPC
6050-6057	8	SERVICE.DNS
6062-6063	2	SERVICE.DNS
3215, 3229, 3223	3	SERVICE.HTTP
5034-5035	2	SERVICE.HTTP
5041, 5043-5045	4	SERVICE.HTTP
5050, 5055, 5071	3	SERVICE.HTTP
5081, 5090, 5123	3	SERVICE.HTTP
5114, 5116-5118	4	SERVICE.HTTP
1100	1	Not applicable. Signature is replaced by 12xx series.
1105-1106	2	Cisco IOS IPS deprecates these signatures, which do not appear in the SDF.
1201-1208	10	OTHER[2] (fragment attack signatures)
3050	2	OTHER 1 (SYN attack signatures)
3150-3152	3	STRING.TCP
4100	1	STRING.UDP
8000	1	Cisco IOS IPS deprecates these signatures, which do not appear in the SDF.

2 The OTHER engine contains existing, hard-coded signatures. Although the standard SDF contains an entry for these signatures, the engine is not dynamically updated. If the SDF that is loaded onto the engine does not contain the signature, the signature is treated as though it has been disabled.

Supported Cisco IOS IPS Signatures in the attack-drop.sdf File

Customers can choose to use Cisco IOS IPS in one of the following ways:

- Download new signatures that are posted on Cisco.com. These signatures can be obtained at the Cisco Intrusion Prevention Alert Center web page. (You must have a valid Cisco.com account to access this web page.)
- Download the attack-drop.sdf file, which contains the signatures that are identified in the table below.

Table 3: Cisco IOS IPS Signatures

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
1006:0	IP options-Strict Source Route	A, D	ATOMIC.IPOPTIONS	Triggers on receipt of an IP datagram in which the IP option list for the datagram includes option 2 (Strict Source Routing).
1102:0	Impossible IP Packet	A, D	ATOMIC.L3.IP	Triggers when an IP packet arrives with source equal to destination address. This signature catches the Land Attack.
1104:0	IP Localhost Source Spoof	A, D	ATOMIC.L3.IP	Triggers when an IP packet with the address of 127.0.0.1, a local host IP address that should never be seen on the network, is detected. This signature can detect the Blaster attack.
1108:0	IP Packet with Proto 11	A, D	ATOMIC.L3.IP	Alarms upon detecting IP traffic with the protocol set to 11. There have been known “backdoors” running on IP protocol 11.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
2154:0	Ping Of Death Attack	A, D	ATOMIC.L3.IP	Triggers when an IP datagram is received with the protocol field in the IP header set to 1 (Internet Control Message Protocol [ICMP]), the Last Fragment bit is set. The IP offset (which represents the starting position of this fragment in the original packet and which is in 8-byte units) plus the rest of the packet is greater than the maximum size for an IP packet.
3038:0	Fragmented NULL TCP Packet	A, D	ATOMIC.TCP	Triggers when a single, fragmented TCP packet with none of the SYN, FIN, ACK, or RST flags set has been sent to a specific host. A reconnaissance sweep of your network may be in progress.
3039:0	Fragmented Orphaned FIN packet	A, D	ATOMIC.TCP	Triggers when a single, fragmented, orphan TCP FIN packet is sent to a privileged port (having a port number less than 1024) on a specific host. A reconnaissance sweep of your network may be in progress.
3040:0	NULL TCP Packet	A, D	ATOMIC.TCP	Triggers when a single TCP packet with none of the SYN, FIN, ACK, or RST flags set has been sent to a specific host. A reconnaissance sweep of your network may be in progress.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
3041:0	SYN/FIN Packet	A, D	ATOMIC.TCP	Triggers when a single TCP packet with the SYN and FIN flags set is sent to a specific host. A reconnaissance sweep of your network may be in progress. The use of this type of packet indicates an attempt to conceal the sweep.
3043:0	Fragmented SYN/FIN Packet	A, D	ATOMIC.TCP	Triggers when a single, fragmented TCP packet with the SYN and FIN flags set is sent to a specific host. A reconnaissance sweep of your network may be in progress. The use of this type of packet indicates an attempt to conceal the sweep.
3129:0	Mimail Virus C Variant File Attachment	A, D, R	SERVICE. SMTP	Fires when an e-mail attachment matching the C Variant of the Mimail virus is detected. The virus sends itself to recipients as the e-mail attachment “photos.zip” that contains the file “photos.jpg.exe” and has “our private photos” in the e-mail subject line. If launched, the virus harvests e-mail addresses and possible mail servers from the infected system.
3140:3	Bagle Virus Activity [4]	A, D, R	SERVICE.HTTP	Fires when HTTP propagation using .jpeg associated with the .Q variant is detected.
3140:4	Bagle Virus Activity [5]	A, D, R	SERVICE.HTTP	Fires when HTTP propagation using .php associated with the .Q variant is detected.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
3300:0	NetBIOS OOB Data	A, D	ATOMIC.TCP	Triggers when an attempt to send out-of-band data to port 139 is detected.
5045:0	WWW xterm display attack	A, D, R	SERVICE.HTTP	Triggers when any cgi-bin script attempts to execute the command xterm -display. An attempt to illegally log in to your system may be in progress.
5047:0	WWW Server Side Include POST attack	A, D, R	SERVICE.HTTP	Triggers when an attempt is made to embed a server side include (SSI) in an http POST command. An attempt to illegally access system resources may be in progress.
5055:0	HTTP Basic Authentication Overflow	A, D	SERVICE.HTTP	A buffer overflow can occur on vulnerable web servers if a very large username and password combination is used with basic authentication.
5071:0	WWW msacds.dll Attack	A, D, R	SERVICE.HTTP	An attempt has been made to execute commands or view secured files, with privileged access. Administrators are highly recommended to check the affected systems to ensure that they have not been illicitly modified.
5081:0	WWW WinNT cmd.exe Access	A, D, R	SERVICE.HTTP	Triggers when the use of the Windows NT cmd.exe is detected in a URL. This signature can detect the NIMDA attack.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
5114: 0 5114:1 5114:2	WWW IIS Unicode Attack	A, D, R	SERVICE.HTTP	Triggers when an attempt to exploit the Unicode ../ directory traversal vulnerability is detected. Looks for the commonly exploited combinations that are included in publicly available exploit scripts. SubSig 2 is know to detect the NIMDA attack.
5126:0	WWW IIS .ida Indexing Service Overflow	A, D, R	SERVICE.HTTP	Alarms if web traffic is detected with the ISAPI extension .ida? and a data size of greater 200 characters.
5159:0	phpMyAdmin Cmd Exec	A, D, R	SERVICE.HTTP	Triggers when access to sql.php with the arguments goto and btnDrop=No is detected.
5184:0	Apache Authentication Module ByPass	A, D, R	SERVICE.HTTP	Fires upon detecting a select statement on the Authorization line of an HTTP header.
5188:0	HTTP Tunneling [6] SubSig 0: GotomyPC	A, D, R	SERVICE.HTTP	Triggers when a computer connects to gotomyPC site.
5188:1	HTTP Tunneling SubSig 1: FireThru	A, D, R	SERVICE.HTTP	Triggers when an attempt to use /cgi-bin/proxy is detected. The /cgi-bin/proxy is used to tunnel connections to other ports using web ports.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
5188:2	HTTP Tunneling SubSig 2: HTTP Port	A, D, R	SERVICE.HTTP	Triggers when a connection is made to exectech-va.com. The site runs a server, which connects to the requested resource and passes the information back to the client on web ports.
5188:3	HTTP Tunneling SubSig 3: httptunnel	A, D, R	SERVICE.HTTP	Triggers when /index/html? is detected on POST request.
5245:0	HTTP 1.1 Chunked Encoding Transfer	A, D, R	SERVICE.HTTP	Fires when HTTP 1.1 chunked encoding transfer activity is detected. This signature is known to detect the Scalper Worm.
5326:0	Root.exe access	A, D, R	SERVICE.HTTP	Alarms upon detecting an HTTP request for root.exe. This signature is known to detect the NIMDA attack.
5329:0	Apache/mod_ssl Worm Probe	A, D, R	SERVICE.HTTP	Fires when a probe by the Apache/mod_ssl worm is detected. If the worm detects a vulnerable web server, a buffer overflow attack is sent to HTTPS port (TCP 443) of the web server. The worm then attempts to propagate itself to the newly infected web server and begins scanning for new hosts to attack.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
5364:0	IIS WebDAV Overflow	A, D, R	SERVICE.HTTP	Fires when a long HTTP request (65000+ characters) is detected with an HTTP header option “Translate:”. An attack to exploit a weakness in the WebDAV component of the IIS web server may be in progress.
5390:0	Swen Worm HTTP Counter Update Attempt	A, D, R	SERVICE.HTTP	Triggers when an attempt to access the URL “/bin/counter.gif?link=bacillus” is detected. A system may be infected by the Swen worm trying the update a counter on a web page located on the server “ww2.fce.vutbr.cz.”
5400:0	Beagle.B (Bagle.B) Web Beacon	A, D, R	SERVICE.HTTP	Fires when a request is made for the script 1.php or 2.php residing on the hosts “www.47df.de” or “www.strato.de,” followed by the argument indicating the trojan’s listening port number, p=8866.
6055:0 6055:1 6055:2	DNS Inverse Query Buffer Overflow	A, D R for subsig 1, 2	SERVICE.DNS	Triggers when an IQUERY request arrives with a data section that is greater than 255 characters.
6056:0 6056:1 6056:2	DNS NXT Buffer Overflow	A, D R for subsig 1, 2	SERVICE.DNS	Triggers when a Domain Name System (DNS) server response arrives with a long NXT resource where the length of the resource data is greater than 2069 bytes or the length of the TCP stream containing the NXT resource is greater than 3000 bytes.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
6057:0 6057:1 6057:2	DNS SIG Buffer Overflow	A, D R for subsig 1, 2	SERVICE.DNS	Triggers when a DNS server response arrives with a long SIG resource where the length of the resource data is greater than 2069 bytes or the length of the TCP stream that contains the SIG resource is greater than 3000 bytes.
6058:0 6058:1	DNS SRV DoS	A, D R for subsig 1	SERVICE.DNS	Alarms when a DNS query type SRV and DNS query class IN is detected with more than ten pointer jumps in the SRV resource record.
6059:0 6059:1 6059:2	DNS TSIG Overflow	A, D R for subsig 2	SERVICE.DNS	Alarms when a DNS query type TSIG is detected and the domain name is greater than 255 characters. This signature is known to detect the Lion work.
6060:0 6060:1 6060:2 6060:3	DNS Complian Overflow	A, D R for subsig 2, 3	SERVICE.DNS	Alarms when a Name Server (NS) record is detected with a domain name greater than 255 characters and the IP address is 0.0.0.0, 255.255.255.255 or a multicast address of the form 224.x.x.x.
6100:0 6100:1	RPC Port Registration	A, D R for subsig 1	SERVICE.RPC	Triggers when attempts are made to register new RPC services on a target host. Port registration is the method used by new services to report their presence to the portmapper and to gain access to a port. Their presence is then advertised by the portmapper.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
6101:0 6101:1	RPC Port Unregistration	A, D R for subsig 1	SERVICE.RPC	Triggers when attempts are made to unregister existing remote procedure call (RPC) services on a target host. Port unregistration is the method used by services to report their absence to the portmapper and to remove themselves from the active port map.
6104:0 6104:1	RPC Set Spoof	A, D R for subsig 1	SERVICE.RPC	Triggers when an RPC set request with a source address of 127.x.x.x is detected.
6105:0 6105:1	RPC Unset Spoof	A, D R for subsig 1	SERVICE.RPC	Triggers when an RPC unset request with a source address of 127.x.x.x is detected.
6188:0	statd dot dot	A, D	SERVICE.RPC	Alarms upon detecting a dot dot slash (../) sequence sent to the statd RPC service.
6189:0 6189:1	statd automount attack	A, D R for subsig 1	SERVICE.RPC	Alarms upon detecting a statd bounce attack on the automount process. This attack targets a vulnerability in the automount process that could be exploited only through localhost.
6190:0 6190:1	statd Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Triggers when a large statd request is sent. This attack could be an attempt to overflow a buffer and gain access to system resources.
6191:0 6191:1	RPC.tooltalk buffer overflow	A, D R for subsig 1	SERVICE.RPC	Fires when an attempt is made to overflow an internal buffer in the tooltalk rpc program.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
6192:0 6192:1	RPC mountd Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Triggers on an attempt to overflow a buffer in the RPC mountd application. This attack may result in unauthorized access to system resources.
6193:0 6193:1	RPC CMSD Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Fires when an attempt is made to overflow an internal buffer in the Calendar Manager Service Daemon, rpc.cmsd.
6194:0 6194:1	sadmind RPC Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Fires when a call to RPC program number 100232 procedure 1 with a UDP packet length greater than 1024 bytes is detected.
6195:0 6195:1	RPC amd Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Detects the exploitation of the RPC AMD Buffer Overflow vulnerability. The trigger for this signature is an RPC call to the berkeley automounter daemons rpc program (300019) procedure 7 that has a UDP length greater than 1024 bytes or a TCP stream length greater than 1024 bytes. The TCP stream length is defined by the contents of the two bytes preceding the RPC header in a TCP packet.
6196:0 6196:1	snmpXdmid Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Fires when an abnormally long call to the RPC program 100249 (snmpXdmid) and procedure 257 is detected.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
6197:0 6197:1	rpc yppaswdd overflow	A, D R for subsig 0	SERVICE.RPC	Fires when an overflow attempt is detected. This alarm looks for an abnormally large argument in the attempt to access yppaswdd.
6276:0 6276:1	TooltalkDB overflow	A, D R for subsig 1	SERVICE.RPC	Alarms upon detecting an RPC connection to rpc program number 100083 using procedure 103 with a buffer greater than 1024.
9200:0	Back Door Response (TCP 12345)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 12345, which is a known trojan port for NetBus as others.
9201:0	Back Door Response (TCP 31337)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 31337, which is a known trojan port for BackFire.
9202:0	Back Door Response (TCP 1524)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 1524, which is a common back door placed on machines by worms and hackers.
9203:0	Back Door Response (TCP 2773)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2773, which is a known trojan port for SubSeven.
9204:0	Back Door Response (TCP 2774)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2774, which is a known trojan port for SubSeven.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
9205:0	Back Door Response (TCP 20034)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 20034, which is a known trojan port for Netbus Pro.
9206:0	Back Door Response (TCP 27374)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 27374, which is a known trojan port for SubSeven.
9207:0	Back Door Response (TCP 1234)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 1234, which is a known trojan port for SubSeven.
9208:0	Back Door Response (TCP 1999)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 1999, which is a known trojan port for SubSeven.
9209:0	Back Door Response (TCP 6711)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6711, which is a known trojan port for SubSeven.
9210:0	Back Door Response (TCP 6712)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6712, which is a known trojan port for SubSeven.
9211:0	Back Door Response (TCP 6713)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6713, which is a known trojan port for SubSeven.
9212:0	Back Door Response (TCP 6776)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6776, which is a known trojan port for SubSeven.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
9213:0	Back Door Response (TCP 16959)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 16959, which is a known trojan port for SubSeven.
9214:0	Back Door Response (TCP 27573)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 27573, which is a known trojan port for SubSeven.
9215:0	Back Door Response (TCP 23432)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 23432, which is a known trojan port for asylum.
9216:0	Back Door Response (TCP 5400)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 5400, which is a known trojan port for back-construction.
9217:0	Back Door Response (TCP 5401)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 5401, which is a known trojan port for back-construction.
9218:0	Back Door Response (TCP 2115)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2115, which is a known trojan port for bugs.
9223:0	Back Door Response (TCP 36794)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 36794, which is a known trojan port for NetBus as well Bugbear.
9224:0	Back Door Response (TCP 10168)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 10168, which is a known trojan port for lovegate.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
9225:0	Back Door Response (TCP 20168)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 20168, which is a known trojan port for lovegate.
9226:0	Back Door Response (TCP 1092)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 1092, which is a known trojan port for lovegate.
9227:0	Back Door Response (TCP 2018)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2018, which is a known trojan port for fizzer.
9228:0	Back Door Response (TCP 2019)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2019, which is a known trojan port for fizzer.
9229:0	Back Door Response (TCP 2020)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2020, which is a known trojan port for fizzer.
9230:0	Back Door Response (TCP 2021)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2021, which is a known trojan port for fizzer.
9231:0	Back Door Response (TCP 6777)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6777, which is a known trojan port for Beagle (Bagle).
9232:0	Back Door Response (TCP 5190)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 5190, which is a known trojan port for the Anig worm.

Signature Id: SubSig ID	Signature Name	Action[3]	SME	Signature Description
9233:0	Back Door Response (TCP 3127)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 3127, which is a known trojan port for the MyDoom.A / Novarg.A virus.
9236:0	Back Door Response (TCP 3128)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 3128, which is a known trojan port for the MyDoom.B / Novarg.B virus.
9237:0	Back Door Response (TCP 8866)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 8866, which is a known trojan port for the Beagle.B (Bagle.B) virus.
9238:0	Back Door Response (TCP 2766)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2766, which is a known trojan port for the DeadHat worm.
9239:0	Back Door Response (TCP 2745)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2745, which is a known trojan port for the Bagle.H-J virus.
9240:0	Back Door Response (TCP 2556)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2556, which is a known trojan port for the Bagle (.M.N.O.P) virus.
9241:0	Back Door Response (TCP 4751)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 4751, which is a known trojan port for the Bagle.U virus.

3 A = alarm, D = drop, R = reset

4 This signature requires port to application mapping (PAM) configuration through the command **ip port-map http port 81**.

5 This signature requires PAM configuration through the command **ip port-map http port 81**.
6 This signature requires PAM configuration through the command **ip port-map http port 8200**.

How to Configure Cisco IOS IPS on a Device

If you want to configure transparent Cisco IOS IPS, you must configure a bridge group before loading IPS onto a device. To configure a bridge group, see the section "Configuring a Bridge Group for Transparent Cisco IOS IPS." If you do not want to configure transparent IPS, skip this task and immediately begin installing IPS onto your device as shown in the tasks below.

Before configuring Cisco IOS IPS on a router, you should determine which one of the following deployment scenarios best addresses your situation and configure the associated task, as appropriate:

- You are loading signatures onto a router through VMS IDS MC or SDM:

  To use VMS IDS MC, see the documents on the VMS index.

  To use SDM, see the chapter "Intrusion Prevention System" in the *Cisco Router and Security Device Manager 2.5 User Guide.*

- You are installing a new router with the latest version of Cisco IOS IPS.

  To perform this task, see the section "Installing Cisco IOS IPS on a New Device."

- Your network is transitioning to Cisco IOS IPS in Cisco IOS Release 12.3(8)T or later.

  To perform this task, see the section "Upgrading to the Latest Cisco IOS IPS Signature Definition File (SDF)."

- You are merging the default (built-in) Cisco IOS IPS signatures with the latest version of the Cisco IOS IPS signature detection file, "attack-drop.sdf."

  To perform this task, see the section "Merging Built-In Signatures with the attack-drop.sdf File."

Configuring Out-of-Order Packet Processing

Prior to Cisco IOS Release 15.2(2)T, Intrusion Prevention System (IPS) sessions use Out-of-Order (OoO) parameters that are configured using the **parameter-map type ooo global** command. For more information, see the section "Configuring OoO Packet Processing Support in the Zone-Based Firewall Applications" in the *Zone-Based Firewall* Configuration Guide.

Perform this task to configure OoO packet processing in Cisco IOS Release 15.2(4)M2.

Note In Cisco IOS Release 15.2(2)T to 15.2(4)M2, you need to enter the **service internal** command in global configuration mode to configure the OoO packet processing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **csdb tcp reassembly max-memory** *memory*
4. **csdb tcp reassembly max-queue-length** *length*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**csdb tcp reassembly max-memory** *memory* **Example:** `Device(config)# csdb tcp reassembly max-memory 12`	Configures the common session database (CSDB) OoO queue memory.
Step 4	**csdb tcp reassembly max-queue-length** *length* **Example:** `Device(config)# csdb tcp reassembly max-queue-length 10`	Configures the CSDB OoO queue length.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Bridge Group for Transparent Cisco IOS IPS

Note You should configure a bridge group only if you want to configure transparent IPS.

- If a BVI is not configured, you must disable IP routing (through the **no ip routing** command) for the bridging operation to take effect.
- If configured, a BVI must be configured with an IP address in the same subnet.

- You *must* configure a BVI if more than two interfaces are placed in a bridge group.
- Bridging between VLAN trunks works only for dot1q encapsulation; Inter-Switch Link (ISL) encapsulation does not work.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **protocol** {**dec** | **ibm** | **ieee** | **vlan-bridge**}
4. **interface** *type number*
5. **bridge-group** *bridge-group*
6. **exit**
7. **bridge irb**
8. **bridge** *bridge-group* **route** *protocol*
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **no shutdown**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *bridge-group* **protocol** {**dec** | **ibm** | **ieee** | **vlan-bridge**} **Example:** `Device(config)# bridge 1 protocol ieee`	Defines the type of Spanning Tree Protocol (STP).
Step 4	**interface** *type number* **Example:** `Device(config)# interface Ethernet0`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**bridge-group** *bridge-group* **Example:** `Device(config-if)# bridge-group 1`	Assigns each network interface to a bridge group. **Note** Complete Step 4 and Step 5 for each interface that you want to assign to a bridge group. **Note** You can also assign subinterfaces to a bridge group to control bridging between VLANs.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 7	**bridge irb** **Example:** `Device(config)# bridge irb`	Enables the Cisco IOS software to route a given protocol between routed interfaces and bridge groups or to route a given protocol between bridge groups. **Note** Step 7 through Step 11 are necessary only if you want to configure a BVI.
Step 8	**bridge** *bridge-group* **route** *protocol* **Example:** `Device(config)# bridge 1 route ip`	Enables the routing of a specified protocol in a specified bridge group.
Step 9	**interface** *type number* **Example:** `Device(config)# interface BVI1`	Configures a BVI and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for an interface.
Step 11	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts a disabled interface.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Examples

The following example shows how to configure interfaces "ethernet0" and "ethernet1" in a bridge group. These interfaces are associated with the BVI interface "BVI1," which can be reached from any host on either of the interfaces through the IP address 10.1.1.1.

```
Device(config)# bridge 1 protocol ieee
Device(config)# interface ethernet0
Device(config-if)# bridge-group 1
Device(config-if)# interface ethernet1
Device(config-if)# bridge-group 1
Device(config-if)# exit
! Configure the BVI.
Device(config)# bridge irb
Device(config)# bridge 1 route ip
Device(config)# interface BVI1
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# no shutdown
```

Troubleshooting Tips

To display the status of each bridge group, use the **show bridge-group** command or to display entries in the bridge table, use the **show bridge** command.

What to Do Next

After you have configured the bridge group, you must configure Cisco IOS IPS as shown in one of the following Cisco IOS IPS tasks, as appropriate to your network needs.

Installing Cisco IOS IPS on a New Device

Perform this task to install the latest Cisco IOS IPS signatures on a device for the first time.

Perform this task to install the default, built-in signatures or the SDF called "attack-drop.sdf"—but not both. If you want to merge the two signature files, you must load the default, built-in signatures as described in this task. Then, you can merge the default signatures with the attack-drop.sdf file as described in the task "Merging Built-In Signatures with the attack-drop.sdf File."

Note Installing the signatures provided in flash is the recommended method in Cisco IOS Release 12.3(8)T for IPS attack mitigation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips sdf location** *url*
4. **ip ips name ips-name** [**list** *acl*]
5. **ip ips signature** *signature-id* [**:sub-signature-id**] {**delete** | **disable** | **list** *acl-list*}
6. **ip ips deny-action ips-interface**
7. **interface** *type number*
8. **ip ips** *ips-name* {**in** | **out**}
9. **end**
10. **show ip ips configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ips sdf location** *url* **Example:** `Device(config)# ip ips sdf location disk2:attack-drop.sdf`	(Optional) Specifies the location in which the device loads the SDF, "attack-drop.sdf." **Note** If this command is not issued, the device loads the default, built-in signatures.
Step 4	**ip ips name ips-name** [**list** *acl*] **Example:** `Device(config)# ip ips name MYIPS`	Creates an IPS rule. **Note** Prior to Cisco IOS Release 12.3(8)T, only standard, numbered ACLs were supported.
Step 5	**ip ips signature** *signature-id* [**:sub-signature-id**] {**delete**	**disable**	**list** *acl-list*} **Example:** `Device(config)# ip ips signature 1000 disable`	(Optional) Attaches a policy to a given signature.

	Command or Action	Purpose
Step 6	**ip ips deny-action ips-interface** **Example:** `Device(config)# ip ips deny-action ips-interface`	(Optional) Creates an ACL filter for the deny actions (denyFlowInline and denyConnectionInline) on the IPS interface rather than the ingress interface. **Note** You should configure this command only if at least one signature is configured to use the supported deny actions, if the input interface is configured for load balancing, and if IPS is configured on the output interface.
Step 7	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1`	Configures an interface type and enters interface configuration mode.
Step 8	**ip ips** *ips-name* {**in** | **out**} **Example:** `Device(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 device prompt is suspended while the signature engines for the newly added or merged signatures are being built. The device 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 seconds. It is recommended that you enable logging messages to monitor the engine building status.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	**show ip ips configuration** **Example:** `Device# show ip ips configuration`	(Optional) Verifies that Cisco IOS IPS is properly configured.

Upgrading to the Latest Cisco IOS IPS Signature Definition File

Perform this task to replace the existing signatures on your router with the latest IPS signature file, attack-drop.sdf.

Note The latest IPS image reads and converts all commands that begin with the words "ip audit" to "ip ips." For example, the **ip audit name** command becomes the **ip ips name** command. Although IPS accepts the **audit** keyword, it generates the **ips** keyword when you show the configuration. Also, if you issue the help character (?), the CLI displays the **ips** keyword instead of the **audit** keyword, and the Tab key used for command completion does not recognize the **audit** keyword.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips name** *ips-name*
4. **ip ips sdf location** *url*
5. **no ip ips location in builtin**
6. **ip ips fail closed**
7. **ip ips deny-action ips-interface**
8. **interface** *type number*
9. **ip ips** *ips-name* {**in** | **out**} [**list** *acl*]
10. **exit**
11. **show ip ips configuration**
12. **show ip ips signatures** [**detailed**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> 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 name** *ips-name* **Example:** `Router(config)# ip ips name MIPS`	Creates an IPS rule.
Step 4	**ip ips sdf location** *url* **Example:** `Router(config)# ip ips sdf location disk2:attack-drop.sdf`	(Optional) Specifies the location where the router loads the SDF. If this command is not issued, the router loads the default SDF.

	Command or Action	Purpose
Step 5	**no ip ips location in builtin** **Example:** `Router(config)# no ip ips location in builtin`	(Optional) Instructs the router not load the built-in signatures if it cannot find the specified signature file. If this command is not issued, the router loads the built-in signatures if the SDF is not found. **Caution** If this command is issued and IPS fails to load the SDF, an error message is received stating that IPS is completely disabled.
Step 6	**ip ips fail closed** **Example:** `Router(config)# ip ips fail closed`	(Optional) Instructs the router to drop all packets until the signature engine is built and ready to scan traffic. If this command is issued, one of the following scenarios occurs: • If IPS fails to load the SDF, all packets are dropped—unless the user specifies an ACL for packets to send to IPS. • If IPS successfully loads the SDF but fails to build a signature engine, all packets that are destined for that engine is dropped. If this command is not issued, all packets are passed without scanning if the signature engine fails to build.
Step 7	**ip ips deny-action ips-interface** **Example:** `Router(config)# ip ips deny-action ips-interface`	(Optional) Creates an ACL filter for the deny actions (denyFlowInline and denyConnectionInline) on the IPS interface rather than the ingress interface. **Note** You should configure this command only if at least one signature is configured to use the supported deny actions, if the input interface is configured for load balancing, and if IPS is configured on the output interface.
Step 8	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet0/1`	Configures an interface type and enters interface configuration mode.
Step 9	**ip ips** *ips-name* {**in** | **out**} [**list** *acl*] **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. • **list** *acl*—Packets that are permitted through a specified ACL are scanned by IPS. **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.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 11	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	(Optional) Displays whether Cisco IOS IPS is properly configured.
Step 12	**show ip ips signatures** [**detailed**] **Example:** `Router# show ip ips signatures`	(Optional) Displays signature configuration, such as signatures that have been disabled.

Merging Built-In Signatures with the attack-drop.sdf File

You may want to merge the built-in signatures with the attack-drop.sdf file if the built-in signatures are not providing your network with adequate protection from security threats. Perform this task to add the SDF and to change default parameters for a specific signature within the SDF or signature engine.

Before you can merge the attack-drop.sdf file with the built-in signatures, you should already have the built-in signatures loaded onto the router as described in "Installing Cisco IOS IPS on a New Device".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip ips location in builtin**
4. **ip ips fail closed**
5. **exit**
6. **copy** [**/erase**] *url* **ips-sdf**
7. **copy ips-sdf** *url*
8. **configure terminal**
9. **ip ips signature** *signature-id* [**:***sub-signature-id*] {**delete** | **disable** | **list** *acl-list*}
10. **ip ips sdf location** *url*
11. **ip ips deny-action ips-interface**
12. **interface** *type name*
13. **ip ips** *ips-name* {**in** | **out**}
14. **end**
15. **show ip ips signatures** [**detailed**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip ips location in builtin** **Example:** `Device(config)# no ip ips location in builtin`	(Optional) Instructs the device not to load the built-in signatures if it cannot find the specified signature file. If this command is not issued, the device loads the built-in signatures if the SDF is not found. **Caution** If this command is issued and IPS fails to load the SDF, an error message is received stating that IPS is completely disabled.
Step 4	**ip ips fail closed** **Example:** `Device(config)# ip ips fail closed`	(Optional) Instructs the device to drop all packets until the signature engine is built and ready to scan traffic. If this command is issued, one of the following scenarios occurs: • If IPS fails to load the SDF, all packets are dropped—unless the user specifies an ACL for packets to send to IPS. • If IPS successfully loads the SDF but fails to build a signature engine, all packets that are destined for that engine is dropped. If this command is not issued, all packets are passed without scanning if the signature engine fails to build.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 6	**copy** [**/erase**] *url* **ips-sdf** **Example:** `Device# copy disk2:attack-drop.sdf ips-sdf`	Loads the SDF in the device. The SDF merges with the signatures that are already loaded in the device, unless the **/erase** keyword is issued. The **/erase** keyword replaces the built-in signatures with the SDF. **Note** The SDF location is not saved in the configuration. The next time the device is reloaded, it refers to a previously specified SDF location in the configuration or it loads the built-in signatures.

	Command or Action	Purpose
		Note Whenever signatures are replaced or merged, the device prompt is suspended while the signature engines for the newly added or merged signatures are being built. The device 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 seconds. It is recommended that you enable logging messages to monitor the engine building status.
Step 7	**copy ips-sdf** *url* **Example:** `Device# copy ips-sdf disk2:my-signatures.sdf`	Saves the SDF that was loaded in the previous step to a specified location. The SDF location is not be saved unless this command is issued.
Step 8	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 9	**ip ips signature** *signature-id* [**:***sub-signature-id*] {**delete** | **disable** | **list** *acl-list*} **Example:** `Device(config)# ip ips signature 1107 disable`	(Optional) Instructs the device to scan for the specified signature but not take any action if the signature is detected. • **list** *acl*—Packets that are permitted through a specified ACL is scanned by IPS.
Step 10	**ip ips sdf location** *url* **Example:** `Device(config)# ip ips sdf location disk2:my-signatures.sdf`	Configures the device to initialize the new SDF.
Step 11	**ip ips deny-action ips-interface** **Example:** `Device(config)# ip ips deny-action ips-interface`	(Optional) Creates an ACL filter for the deny actions (denyFlowInline and denyConnectionInline) on the IPS interface rather than the ingress interface. **Note** You should configure this command only if at least one signature is configured to use the supported deny actions, if the input interface is configured for load balancing, and if IPS is configured on the output interface.
Step 12	**interface** *type name* **Example:** `Device(config)# interface GigabitEthernet0/1`	Configures an interface type and enters interface configuration mode.
Step 13	**ip ips** *ips-name* {**in** | **out**} **Example:** `Device(config-if)# ip ips MYIPS in`	Applies an IPS rule at an interface and reloads the device and reinitializes Cisco IOS IPS.

	Command or Action	Purpose
		Note The device prompt disappears while the signatures are loading and the signature engines are building. The device prompt reappears after the signatures have been loaded and the signature engines have been built.
Step 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode.
Step 15	**show ip ips signatures** [**detailed**] **Example:** `Device# show ip ips signatures`	(Optional) Verifies signature configuration, such as signatures that have been disabled or marked for deletion.

Monitoring Cisco IOS IPS Signatures Through Syslog Messages or SDEE

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. **exit**
7. **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.

	Command or Action	Purpose
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.
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	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**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.

Troubleshooting Cisco IOS IPS

Interpreting Cisco IOS IPS System Messages

Table 4: Cisco IOS IPS System Alarm, Status, and Error Messages

System Message	Description
Alarm Messages	
`%IPS-4-SIGNATURE:Sig:1107 Subsig:0 Sev:2 RFC1918 address [192.168.121.1:137 -> 192.168.121.255:137]`	An IPS signature has been triggered.
`%IPS-5-SIGNATURE:Sig:1107 Subsig:0 Global Summary:50 alarms in this interval`	A flood of the specified IPS signature has been seen and summarized. (For example, signature 1107 has been seen 50 times.)
Status Messages	
`%IPS-6-ENGINE_READY:SERVICE.HTTP - 183136 ms - packets for this engine will be scanned`	An IPS signature engine has been built and is ready to scan packets.
`%IPS-6-ENGINE_BUILD_SKIPPEf:STRING.UDP - there are no new signature definitions for this engine`	There are not any signature definitions or changes to the existing signature definitions of an IPS signature engine, and the engine does not have to be rebuilt.
`%IPS-5-PACKET_DROP:SERVICE.DNS - packets dropped while engine is building`	Packets are being dropped because the specified IPS module is not functioning and the **ip ips fail closed** command is configured. The message is rate limited to 1 message per 60 seconds.
`%IPS-5-PACKET_UNSCANNEf:SERVICE.DNS - packets passed unscanned while engine is building`	Packets are passing through the network but are not being scanned because the specified IPS module is not functioning and the **ip ips fail closed** command is not configured. The message is rate limited to 1 message per 60 seconds.
`%IPS-6-SDF_LOAD_SUCCESS:SDF loaded successfully from flash:sdf_8http.xml`	An SDF is successfully loaded from a given location.

System Message	Description
Error Messages	

`%IPS-3-BUILTIN_SIGS:Configured to load builtin signatures`
`%IPS-3-BUILTIN_SIGS:Not Configured to load builtin signatures`
`%IPS-3-BUILTIN_SIGS:Failed to load builtin signatures`
	One of these three messages can be displayed when IPS loads the built-in signatures.

`%IPS-5-ENGINE_UNKNOWN: SERVICE.GENERIC - unknown engine encountered while parsing SDF`	The router has encountered an unknown and unsupported signature engine while parsing the SDF. To prevent this message from being generated again, ensure that the SDF being loaded on the router does not contain any engines that are not supported by IPS.
`%IPS-5-UNSUPPORTED_PARAM: SERVICE.RPC 6275:1 isSweep=False - bad parameter - removing parameter`	The router has encountered an unsupported parameter while parsing the SDF. The signature is deleted if the unsupported parameter is required for the signature. The parameter is removed from the signature if it is not required. To prevent this message from being generated again, ensure that the SDF being loaded on the router does not contain any parameters that are not supported by IPS.
`%IPS-3-ENGINE_BUILD_FAILEf: SERVICE.HTTP - 158560 ms - engine build`	One of the signature engines fails to build after an SDF is loaded. A message is sent for each engine that fails. An engine typically fails to build because of low memory, so increasing router memory can alleviate the problem. Also, try to load the SDF immediately after a route reboots, which is when system resources are available.
`%IPS-4-SDF_PARSE_FAILEf: not well-formed (invalid token) at Line 1 Col 0 Byte 0 Len 1006`	An SDF has not parsed correctly. The SDF might have been corrupt.
`%IPS-4-SDF_LOAD_FAILEf: failed to parse SDF from tftp://tftp-server/sdf.xml`	An SDF fails to load. The SDF may fail for any of the following reasons: • Fails to load if it resides on a network server that cannot be reached • Does not have the correct read permissions
`%IPS-2-DISABLEf: IPS removed from all interfaces - IPS disabled`	IPS has been disabled. This message indicates why IPS has been disabled.

Conditions of an SME Build Failure

Sometimes an SME that is being built fails. The SME can fail because it is attempting to load a corrupted SDF file or it exceeds memory limitations of the router. If a failure occurs, Cisco IOS IPS is designed to handle it. Possible failures are as follows:

- By default, IPS is designed to "fail open," which means that if an SME does not build, all packets that are destined for that particular engine passes traffic without scanning.
- If IPS cannot load the attack-drop.sdf file onto a router, the router reverts to the previously loaded available signatures. (In most cases, the previously loaded signatures are the Cisco IOS built-in signatures.)
- If an engine build fails when you are merging the attack-drop.sdf file with the built-in signatures, IPS reverts, by default, to the previously available engine (or engines).

The default behavior for engine failure allows for packets to be passed unscanned. To prevent traffic from being passed unscanned, issue the **ip ips fail closed** command, which forces the router to drop all packets if an SME build fails.

Note If a signature or a signature parameter is not supported, Cisco IOS prints a syslog message, indicating that the signature or parameter is not supported.

Configuration Examples

Example: Configuring Out-of-Order Packet Processing

```
Device# configure terminal
Device(config)# csdb tcp reassembly max-memory 12
Device(config)# csdb tcp reassembly max-queue-length 10
Device(config)# end
```

Example: Loading the Default Signatures

The following example shows how to specify the Cisco IOS IPS commands to load the default, built-in signatures. Note that a configuration option for specifying an SDF location is not necessary; built-in signatures reside statically in Cisco IOS software.

```
!
ip ips po max-events 100
ip ips name MYIPS
!
interface GigabitEthernet0/1
 ip address 10.1.1.16 255.255.255.0
 ip ips MYIPS in
 duplex full
 speed 100
 media-type rj45
```

```
 no negotiation auto
!
```

Example: Loading the attack-drop.sdf File

The following example shows the basic configuration necessary to load the attack-drop.sdf file onto a router running Cisco IOS IPS. Note that the configuration is almost the same as loading the default signatures onto a router, except for the **ip ips sdf location** command, which specifies the attack-drop.sdf file.

```
!
ip ips sdf location disk2:attack-drop.sdf
ip ips name MYIPS
!
interface GigabitEthernet0/1
 ip address 10.1.1.16 255.255.255.0
 ip ips MYIPS in
 duplex full
 speed 100
 media-type rj45
 no negotiation auto
!
```

Example: Merging the attack-drop.sdf File with the Default Built-in Signatures

The following example shows how to configure the router to load and merge the attack-drop.sdf file with the default signatures. After you have merged the two files, it is recommended that you copy the newly merged signatures to a separate file. The router can then be reloaded (through the **reload** command) or reinitalized to recognize the newly merged file (as shown the following example).

```
!
ip ips name MYIPS
!
interface GigabitEthernet0/1
 ip address 10.1.1.16 255.255.255.0
 ip ips MYIPS in
 duplex full
 speed 100
 media-type rj45
 no negotiation auto
!
!
! Merge the flash-based SDF (attack-drop.sdf) with the built-in signatures.
Router# copy disk2:attack-drop.sdf ips-sdf
! Save the newly merged signatures to a separate file.
Router# copy ips-sdf disk2:my-signatures.sdf
!
! Configure the router to use the new file, my-signatures.sdf
Router# configure terminal
Router(config)# ip ips sdf location disk2:my-signatures.sdf
! Reinitialize the IPS by removing the IPS rule set and reapplying the rule set.
Router(config-if)# interface gig 0/1
Router(config-if)# no ip ips MYIPS in
!
*Apr 8 14:05:38.243:%IPS-2-DISABLEf:IPS removed from all interfaces - IPS disabled
!
Router(config-if)# ip ips MYIPS in
!
Router(config-if)# exit
```

Additional References Cisco IOS Intrusion Prevention System

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

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 Configuring Cisco IOS IPS

The following table provides release information about the feature or features described in this module. This table 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.

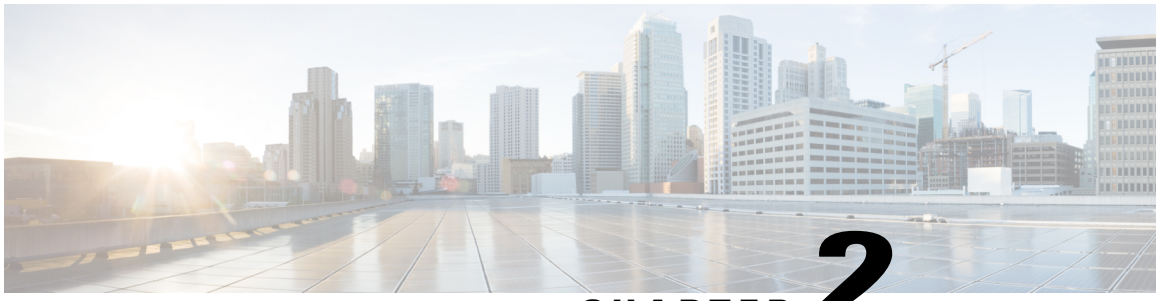
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for Configuring Cisco IOS IPS

Feature Name	Software Releases	Feature Configuration Information
Cisco IOS Intrusion Prevention System (IPS)	12.3(8)T 12.3(14)T	

Feature Name	Software Releases	Feature Configuration Information
		The Cisco IOS Intrusion Prevention System (IPS) uses the following methods to protect a network from internal and external attacks and threats: • IPS signatures are dynamically updated and posted to Cisco.com on a regular basis so that customers can access signatures that help protect their network from the latest known network attacks. • A Parallel Signature Scanning Engine is used to scan for multiple patterns within a signature microengine (SME) at any given time. IPS signatures are no longer scanned on a serial basis. • Cisco IOS IPS supports both named and numbered extended access control lists (ACLs). The following commands were introduced by this feature: **clear ip sdee**, **copy ips-sdf**, **debug ip ips**, **debug ip sdee**, **ip ips fail closed**, **ip ips sdf location**, **ip sdee events**, **ip sdee subscriptions**, **no ip ips sdf builtin**, **show ip sdee.** • Supports access to more recent virus and attack signatures with the addition of three more SMEs—STRING.TCP, STRING.ICMP, and STRING.UDP. • Intelligent and local shunning is supported, which allows Cisco IOS IPS to shun offending traffic on the same router that Cisco IOS IPS is configured.

Feature Name	Software Releases	Feature Configuration Information
		• The **ip ips deny-action ips-interface** command was added, which allows users to choose between two available ACL filter settings for detecting offending packets. Support for the Post Office Protocol was deprecated and the following commands were removed from the Cisco IOS software: **ip ips po local**, **ip ips po max-events**, **ip ips po protected**, and **ip ips po remote**.
Transparent Cisco IOS IPS	12.4(2)T	Support was added for Layer 2 transparent bridging for Cisco IOS IPS. Transparent Cisco IOS IPS eases certain network and management deployment by allowing users to “drop” a device running Cisco IOS IPS in front of their existing network without changing the statically defined IP addresses of their network-connected devices. Thus, users can allow selected devices from a subnet to traverse the IPS while access to other devices on the same subnet is denied. No commands were introduced or modified for this feature.



CHAPTER 2

IOS IPS Auto Update Functionality

Cisco provides IOS Intrusion Prevention System (IPS) software and signature updates on a regular basis. The IOS IPS Auto Update feature does a periodic update of these signatures automatically. In Cisco IOS Release 15.5(2)T and later releases, the auto update is provided by the BSD infrastructure. Prior to this release, the auto update was done by the IDA application.

This module provides an overview of the feature and explains how to configure it.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 table.

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

Information About IOS IPS Auto Update Functionality

IOS IPS Auto Update Overview

Cisco IOS Intrusion Prevention System (IPS) protects a network infrastructure from malicious traffic or attacks. Cisco provides IOS IPS software and signature updates on a regular basis. As new forms of network

attacks are devised, new signatures are developed to combat them. IOS IPS auto update does a periodic update of these signatures automatically.

In Cisco IOS Release 15.5(2)T and later releases, IOS IPS auto update uses the Borderless Software Delivery (BSD) infrastructure. IOS IPS auto update will only support update requests coming through BSD. Prior to this release, IDA was used for auto update.

IOS IPS auto update supports two kinds of auto updates and these are:

- Auto update from a local FTP/TFTP server:

  You can configure IOS IPS to automatically update its signatures from a local URL (using FTP/TFTP). You need to manually download the signature file from Cisco.com and place it in the FTP/TFTP server path which is configured in IOS IPS. Based on the configuration, IOS IPS periodically updates its signatures from the local server path.

  Note: Auto update from local a local URL does not verify if the signature file is the latest or not; but takes the signature file that is available in the configured location.

- Auto update from www.cisco.com:

  You can configure IOS IPS to automatically update its signatures from Cisco.com. IOS IPS checks for the latest signature package availability, and if an upgrade to the currently running signature version is available, the signature is downloaded and upgraded.

Catalog File Service Functionality

Borderless Software Delivery (BSD) server provides the catalog file service functionality to support selective IOS IPS image update.

A catalog which consists of filters corresponding to image versions and packages which are supported for these image versions are uploaded on the BSD server. When the IOS IPS sends a request through the BSD client, the server sends a response that contains the list of software updates available for the image version running on the router. The IOS IPS interface selects the software update to be retrieved from the BSD server, and downloads the image. Image download from Cisco.com is done using HTTP.

You can configure the interval at which to initiate the auto update. IOS IPS starts a timer based on the configured interval, and on expiry of the timer the auto update starts.

How to Configure IOS IPS Auto Update Functionality

Configuring IOS IPS Auto Update

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. **exit**
8. **bsd-client server url** *url*
9. **password encryption aes**
10. **key config-key password-encryption**
11. **exit**
12. **show ip ips configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ips auto-update** **Example:** `Device(config)# ip ips auto-update`	Enables automatic signature updates for Cisco IOS Intrusion Prevention System (IPS) and enters IPS-auto-update configuration mode.
Step 4	**cisco** **Example:** `Device(config-ips-auto-update)# cisco`	Enables automatic IOS IPS signature updates from Cisco.com.

	Command or Action	Purpose
Step 5	**occur-at** [**monthly** | **weekly**] *days minutes hours* **Example:** `Device(config-ips-auto-update)# occur-at weekly 4 23 23`	Defines a preset time after which IOS IPS automatically obtains updated signature information.
Step 6	**username** *name* **password** *password* **Example:** `Device(config-ips-auto-update)# username myips password secret`	Defines a username and password to access signature files from the server.
Step 7	**exit** **Example:** `Device(config-ips-auto-update)# exit`	Exits IPS-auto-update configuration mode and returns to global configuration mode.
Step 8	**bsd-client server url** *url* **Example:** `Device(config)# bsd-client server url https://cloudsso.cisco.com/as/token.oauth2`	Configures the Borderless Software Delivery (BSD) server URL to auto download signatures.
Step 9	**password encryption aes** **Example:** `Device(config)# password encryption aes`	Enables a type 6 encrypted preshared key.
Step 10	**key config-key password-encryption** **Example:** `Device(config)# key config-key password-encryption`	Stores a type 6 encryption key in local NVRAM.
Step 11	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 12	**show ip ips configuration** **Example:** `Device# show ip ips configuration`	Displays IPS information such as configured sessions, signatures, and additional configuration information that includes default values.

Example

The following is sample output from the **show ip ips configuration** command:

```
Device# show ip ips configuration

Event notification through syslog is enabled
Event notification through Net Director is enabled
Default action(s) for info signatures is alarm
Default action(s) for attack signatures is alarm
Default threshold of recipients for spam signature is 25
PostOffice:HostID:5 OrgID:100 Addr:10.2.7.3 Msg dropped:0
```

```
HID:1000 OID:100 S:218 A:3 H:14092 HA:7118 DA:0 R:0
    CID:1 IP:172.16.0.20 P:45000 S:ESTAB (Curr Conn)

Audit Rule Configuration
 Audit name AUDIT.1
    info actions alarm
```

Configuration Examples for IOS IPS Auto Update Functionality

Verifying IOS IPS Auto Update Functionality

Use the following commands to verify your IOS IPS auto update functionality:

SUMMARY STEPS

1. **enable**
2. **show ip ips auto-update**
3. **show ip ips statistics**
4. **clear ip ips statistics**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted

Step 2 **show ip ips auto-update**

Example:

Displays the automatic signature update configuration.

```
Device# show ip ips auto-update
```

Step 3 **show ip ips statistics**

Example:

Displays the information such as the number of packets audited and the number of alarms sent.

```
Device# show ip ips statistics
```

Step 4 **clear ip ips statistics**

Example:

Resets statistics of packets analyzed and alarms sent.

```
Device# clear ip ips statistics
```

Example: Configuring IOS IPS Auto Update

```
Device# configure terminal
Device(config)# ip ips auto-update
Device(config-ips-auto-update)# cisco
Device(config-ips-auto-update)# occur-at weekly 4 23 23
Device(config-ips-auto-update)# username myips password secret
Device(config-ips-auto-update)# exit
Device(config)# bsd-client server https://cloudsso.cisco.com/as/token.oauth2
Device(config)# password encryption aes
Device(config)# key config-key password-encryption
Device(config)# end
```

Additional References for IOS IPS Auto Update Functionality

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

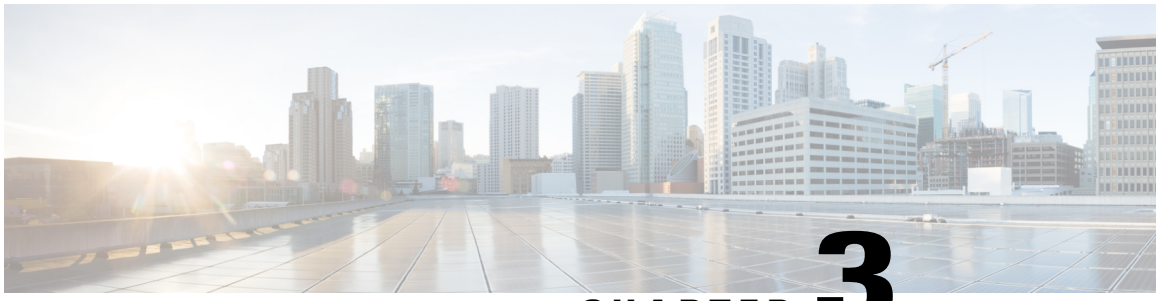
Feature Information for IOS IPS Auto Update Functionality

The following table provides release information about the feature or features described in this module. This table 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.

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

Table 6: Feature Information for IOS IPS Auto Update Functionality

Feature Name	Software Releases	Feature Configuration Information
IOS IPS Auto Update Functionality	15.5(2)T	Cisco provides IOS Intrusion Prevention System (IPS) software and signature updates on a regular basis. The IOS IPS Auto Update feature does a periodic update of these signatures automatically. In Cisco IOS Release 15.5(2)T and later releases, the auto update is provided by the BSD infrastructure. Prior to this release, the auto update was done by the IDA application. The following commands were introduced or modified for this feature: **bsd-client server**, **clear ip ips statistics**, **ips signature update**, **show ip ips auto-update**, **show ip ips statistics**.



CHAPTER 3

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

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 caveats and feature information, see Bug Search Tool and 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 table.

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

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

System and Image Requirements for Cisco IOS IPS 5.x

- For Cisco IOS Intrusion Prevention System (IPS) signatures, see the "Cisco IOS Signature Package" section for more information.
- Cisco IOS IPS system requirements depend on the type of deployment, bandwidth requirements, and security requirements. The larger the number of signatures, the larger the amount of memory consumed.
- You must generate a Rivest, Shamir and Adleman (RSA) crypto key and load the public signature on your device for signature decryption.

This following Cisco public key configuration can be cut and pasted directly into your device 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 (realm-cisco.pub.key.txt) 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 device 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 the 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 CLI and features.

When reconfiguring Cisco IOS IPS on a device 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 is 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.

```
Device# 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 Cisco.com Capability in IOS IPS Support

A device must have access to Cisco.com to upgrade IPS signatures directly from Cisco.com. If the device does not have access to Cisco.com, signature file updates can be retrieved from a local server.

Some devices that are configured for IPS signature autoupdate may not have the necessary certificate trustpoints defined to support HTTPS communications to Cisco.com. As a workaround for this issue, you can do the following:

- Open the URLhttps://www.cisco.com/cgi-bin/front.x/ida/locator/locator.pl in your browser.
- Download and save the SSL certificate chain as Base-64 encoded X.509 .cer files.
- Manually define and authenticate trustpoints for the root and sub root, and identity the certificate on the device.

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

Danger Do not enable all Intrusion Prevention System (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.

- Cisco IOS IPS 5.x format signatures are not backward compatible with Cisco IOS IPS 4.x signature definition files (SDFs).
- Automatic signature updates from Cisco.com is not available for Cisco routers running the Cisco IOS Releases 12.4(24)T, and 15.0(1)M to 15.5(1)T. As a workaround, you can do the following:
  - Open a TAC case to have an updated signature file published to a valid Cisco.com ID.
  - Manually download the signature package from Cisco.com, and either manually apply the signature to the router or configure automatic updates from an internal file server.

In Cisco IOS Release 15.5(2)T, automatic signature updates are available through Borderless Software Delivery (BSD) infrastructure.

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 compared 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.
- 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 Package

The latest Cisco IOS IPS signature package can be accessed from Cisco.com using the following URL: http://www.cisco.com/cisco/software/release.html?mdfid=281442967&release=S636&relind=AVAILABLE&flowid=4836&softwareid=280775022&rellifecycle=&reltype=latest

Note Ensure that you have your Cisco userid, and password to access this URL.

- The IOS-Sxxx-CLI.pkg file is listed on this web page. See the "Configuring Cisco IOS IPS on Your Router" for more information on creating a directory on the router flash directory where the required signature files and configurations are stored. Alternatively, you can use a Cisco USB flash drive connected to the router's USB port to store the signature files and configurations. The USB flash drive needs to remain connected to the router USB port if it is used as the IOS IPS configuration directory location. IOS IPS also supports any IOS File System as its configuration location with proper write access.

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 supportef:

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 "Tuning Signatures per Signature ID" for more information about the configuration of this feature.

Signature Auto Update Configuration

IPS signature auto updates for your router can be configured in one of the following ways:

- Manually configuring signature updates from Cisco.com. An administrator can use the CLI to specify, download and upgrade to a new signature package posted for the IOS directly from Cisco.com.
- Configuring the automatic signature updates to be initiated from a local file server.
- Configuring signature updates to be automatically updated from Cisco.com allows the 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. 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.

Note The manual and automatic updating of IPS signatures was introduced through the Direct Download from CCO capability in IOS IPS feature in Cisco IOS Release 15.1(1)T. This feature was unavailable prior to this release.

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

Retiring All Signatures and Selecting a Category of Signatures

Device memory and resource constraints prevent a device from loading all Cisco IOS IPS signatures. We recommend 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 device to load information for all signatures, but the device does not build the parallel scanning data structure.

Note If you do not adhere to the recommendation about retiring signatures, your device may crash based on its resource constraints.

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 device 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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ips signature-category** **Example:** `Device(config)# ip ips signature-category`	Enters enters IPS category configuration mode.
Step 4	**category** *category* [*subcategory*] **Example:** `Device(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:** `Device(config-ips-category-action)# retired true`	Specifies that the device 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:** `Device(config-ips-category-action)# exit`	Exits IPS category action configuration mode.
Step 7	**category** *category* [*subcategory*] **Example:** `Device(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:** `Device(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	**end** **Example:** `Device(config-ips-category-action)# end`	Exits IPS category action and IPS category configuration modes and returns to privileged EXEC mode.

What to Do Next

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

Configuring Cisco IOS IPS on Your Device

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 device 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. **end**
9. **show ip ips configuration**
10. **show ip ips signature** *count*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**mkdir flash:/ips5** **Example:** `Device# 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.

	Command or Action	Purpose
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip ips name** *ips-name* **Example:** `Device(config)# ip ips name myips`	Creates an IPS rule.
Step 5	**ip ips config location** *url* **Example:** `Device(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:** `Device(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:** `Device(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 device prompt is suspended while the signature engines for the newly added or merged signatures are being built. The device 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.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 9	**show ip ips configuration** **Example:** `Device# show ip ips configuration`	(Optional) Verifies that Cisco IOS IPS is properly configured.

	Command or Action	Purpose
Step 10	**show ip ips signature** *count* **Example:** `Device# 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:

```
Device# 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
Enablef: 7
Retiref: 2
Compilef: 5
Signature Micro-Engine: service-http
Total Signatures: 541
Enablef: 284
Retiref: 336
Compilef: 205
Signature Micro-Engine: string-tcp
Total Signatures: 487
Enablef: 332
Retiref: 352
Compilef: 135
Signature Micro-Engine: string-udp
Total Signatures: 50
Enablef: 3
Retiref: 23
Compilef: 27
Signature Micro-Engine: state
Total Signatures: 26
Enablef: 15
Retiref: 23
Compilef: 3
Signature Micro-Engine: atomic-ip
Total Signatures: 140
Enablef: 87
Retiref: 93
Compilef: 46
Inactive - invalid params: 1
Signature Micro-Engine: string-icmp
Total Signatures: 2
Enablef: 0
Retiref: 1
Compilef: 1
Signature Micro-Engine: service-ftp
Total Signatures: 3
Enablef: 3
Compilef: 3
Signature Micro-Engine: service-rpc (INACTIVE)
Signature Micro-Engine: service-dns
Total Signatures: 1
Enablef: 1
Retiref: 1
Signature Micro-Engine: normalizer
Total Signatures: 9
Enablef: 9
Compilef: 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 "Configuring Cisco IOS IPS on Your Router" before loading new signatures.

Flexible Signatures Ordered and Incremental

Each signature is compiled 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. **end**
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.

	Command or Action	Purpose
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	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode.
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.

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:

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 for a 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**
4. **ip ips signature-definition**
5. **signature** *signature-id* [*signature-id*]
6. **engine**
7. **event-action** *action*
8. **exit**
9. **alert-severity** {**high** | **medium** | **low** | **informational**}
10. **fidelity-rating** *rating*
11. **status**
12. **enabled** {**true** | **false** }
13. **exit**
14. **show ip ips signature**
15. **show ip ips sig-clidelta**

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.

	Command or Action	Purpose
Step 4	**ip ips signature-definition** **Example:** `Router(config)# ip ips signature-definition`	Enters signature-definition-signature configuration mode.
Step 5	**signature** *signature-id* [*signature-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.
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.

	Command or Action	Purpose
Step 10	**fidelity-rating** *rating* **Example:** `Router(config-sigdef-action)# fidelity-rating 2`	(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.
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 "Retiring All Signatures and Selecting a Category of Signatures" 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. **end**
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 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**

	Command or Action	Purpose
		• **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.
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	**end** **Example:** `Router(config-ips-category-action)# end` **Example:**	Returns to EXEC mode.
Step 11	**show ip ips signature** **Example:** `Router# show ip ips signature`	(Optional) Verifies the signature category changes that have been made.

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 to rate how important the system is to the network. • The **target-address** keyword and arguments specify a host, which can consist of a single IP address or range of IP addresses.
Step 5	**exit** **Example:** `Router(config-rul)# exit`	Exits config-rule configuration mode.

Configuring Signature Auto Updates

Configuring Signature Auto Updates from a Local Server

Note This functionality was introduced through the Cisco IOS IPS 5.x Signature Format and Usability Enhancements feature in Cisco IOS Release 12.4(11)T.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key config-key password-encrypt**
4. **password encryption aes**
5. **ip ips auto-update**
6. **occur-at** {[**monthly** | **weekly**] *days minutes hours*}
7. **username** *name* **password** *password*
8. **url** *url*
9. **exit**
10. **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.
Step 3	**key config-key password-encrypt** **Example:** `Router(config-ips-auto-update)# key config-key password-encrypt`	Stores a type 6 encryption key in private NVRAM.

	Command or Action	Purpose
Step 4	**password encryption aes** **Example:** `Router(config-ips-auto-update)# password encryption aes`	Enables a type 6 encrypted preshared key. **Note** Once the **key config-key password-encrypt** and **password encryption aes** commands are configured, they enable the password (symmetric cipher Advanced Encryption Standard (AES) encrypts the keys).
Step 5	**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 6	**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 7	**username** *name* **password** *password* **Example:** `Router(config-ips-auto-update)# username myips password secret`	(Optional) Defines a username and password for the signature update function.
Step 8	**url** *url* **Example:** `Router(config-ips-auto-update)# url tftp://10.10.10.5/username1/ips-auto-update/`	URL from the local server on which the router retrieves the Cisco IOS IPS signature configuration files.

Step 9	**exit**
	
	Example:
	
	`Router(config-ips-auto-update)# exit`
	`Router(config)# exit`
		Exits IPS auto-update and global configuration modes.

Step 10	**show ip ips auto-update** **Example:** `Router# show ip ips auto-update`	Verifies the signature update configuration.

Preparing SSL Certificates for Cisco.com Signature Auto Updates

In order for IPS signatures to be automatically upgraded from Cisco.com, the current SSL certificate(s) need to be retrieved and manually installed on the router. Cisco update servers use a mix of Verisign and CyberTrust certificates, so the specific certificate may vary depending on the origin of the update connection. Although the steps are the same, examples in the following configuration show the Verisign certificate chain (which is the most commonly used).

Note SSL certificates are typically valid for a 12 month period. Ensure that this task is repeated periodically to refresh the installed certificates.

Note The examples in this task use Internet Explorer browser. The certificate export process may be different if you are using a different browser.

SUMMARY STEPS

1. Open your browser and go to the Cisco IDA-server URL.
2. Enter your Cisco user ID and password.
3. To display the website SSL certificate, click on the padlock icon to the right of the URL field and select **View Certificates** from the drop-down menu.
4. In the Certificate pop-up window, click on the Certification Path tab to view the certification path.
5. To manually export the root and sub-root certificates in the chain, highlight each level individually and then click on the **View Certificate** button.
6. In the new Certificate pop-up window, click on the **Details** tab and click on the **Copy to File** button.
7. Click **Next** in the Certificate Export Wizard pop-up window. Select the **Base-64 encoded X.509 (.CER)** format and click **Next**. Specify a meaningful filename and export location. For this example, you can save the files **root** and **sub-root** locally to the desktop. Repeat steps 5 through 7 for any sub-root servers in the chain.

DETAILED STEPS

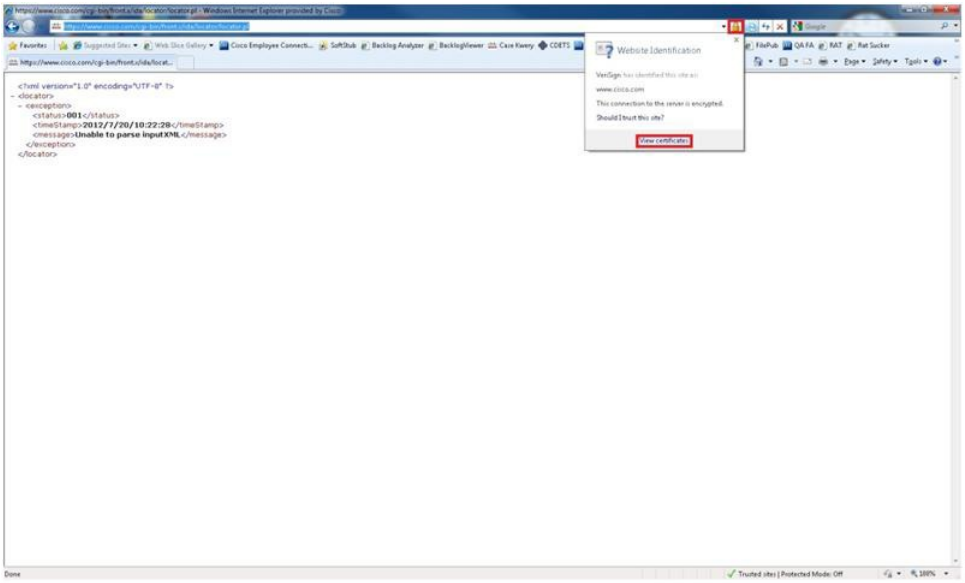
Step 1 Open your browser and go to the Cisco IDA-server URL.

Example:
https://www.cisco.com/cgi-bin/front.x/ida/locator/locator.pl

Step 2 Enter your Cisco user ID and password.

Step 3 To display the website SSL certificate, click on the padlock icon to the right of the URL field and select **View Certificates** from the drop-down menu.

Example:



The Certificate pop-up window displays.

Step 4 In the Certificate pop-up window, click on the Certification Path tab to view the certification path.

Example:

Note The identity certificate for www.cisco.com is signed by VeriSign root and sub-root CA servers in the example above.

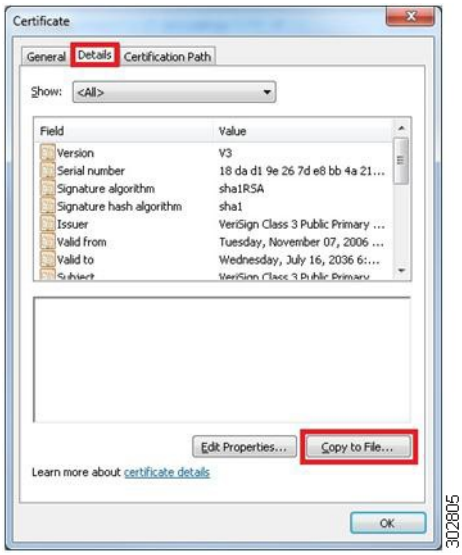
Step 5 To manually export the root and sub-root certificates in the chain, highlight each level individually and then click on the **View Certificate** button.

Example:

A new Certificate pop-up window displays.

Step 6 In the new Certificate pop-up window, click on the **Details** tab and click on the **Copy to File** button.

Example:



The Certificate Export Wizard pop-up window displays.

Step 7 Click **Next** in the Certificate Export Wizard pop-up window. Select the **Base-64 encoded X.509 (.CER)** format and click **Next**. Specify a meaningful filename and export location. For this example, you can save the files **root** and **sub-root** locally to the desktop. Repeat steps 5 through 7 for any sub-root servers in the chain.

What to Do Next

Now you are ready to configure the router so that the upgrading of IPS signatures can be configured automatically from Cisco.com. See the “Upgrading Signatures Automatically from Cisco.com” section for more information.

Creating a PKI Trustpoint for Auto Signature Updates

Use this task to create PKI trustpoints, which are required for manual and automatic signature updates from Cisco.com.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment terminal** [**pem**]
5. **revocation-check none**
6. **exit**
7. **crypto pki authenticate** *name*

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	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint root`	Creates a trustpoint and enters ca-trustpoint configuration mode.
Step 4	**enrollment terminal** [**pem**] **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specifies the manual cut-and-paste certificate enrollment. • **pem** -- (Optional) Adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 5	**revocation-check none** **Example:** `Router(ca-trustpoint)# revocation-check none`	Specifies that certificate checking of the revocation status is not required.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and enters global configuration mode.

Step 7	**crypto pki authenticate** *name*
	
	Example:
	
	`Router(config)# crypto pki authenticate root`
	
	`Enter the base 64 encoded CA certificate.`
	`End with a blank`
		Authenticates the CA server (by getting the certificate of the CA server).
		
		• *name* -- The name of the CA.
		
		Once you enter the name of the CA, you are prompted to enter the base 64 encoded CA certificate that is in the .CER file that you created in the "Preparing SSL Certificates for Cisco.com Signature Auto Updates" section. To view the certificate in the .CER file, open the .CER file in a text editor such as Wordpad. Copy the certificate details directly into the terminal window. Once pasted, hit "Enter" to go to a new line and then type **quit** to exit. Type **yes** to accept the certificate.

Certificate Installation Examples

In the following example, "root" is defined for the first trustpoint CA:

In the following example "sub-root" is defined for the second trustpoint CA:

Manually Configuring Signature Auto Updates from Cisco.com

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

Note This functionality was introduced through the Direct Download from CCO capability in IOS IPS feature in Cisco IOS Release 15.1(1)T.

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. **configure terminal**
3. **ida-client server url** *url*
4. **password encryption aes**
5. **key config-key password-encrypt**
6. **exit**
7. **ips signature update cisco** [**version** {**next** | **latest** | *release-version*} | **username** *name* **password** *password*]
8. **show ip ips configuration**
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.
Step 3	**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 4	**password encryption aes** **Example:** `Router(config)# password encryption aes`	Enables the encryption of the password with a type 6 encrypted preshared key.
Step 5	**key config-key password-encrypt** **Example:** `Router(config)# key config-key password-encrypt`	Configures the encrypted preshared key that is used to encrypt all other keys in the router.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Step 7	**ips signature update cisco** [**version** {**next** | **latest** | *release-version*} | **username** *name* **password** *password*]
	
	Example:
	`Router# ips signature update cisco`
	
	Example:
	`Router# ips signature update cisco username myips password secret`
		Initiates a one-time download of an IPS signatures (signature package) from Cisco.com.
		(Optional) The **version** keyword with the **next** keyword specifies the next signature file package version from the current signature file on the router.
		(Optional) The **version** keyword with the with the **latest** keyword specifies the IOS IPS to search for the latest signature package.
		(Optional) The **version** keyword with the *signature* argument specifies a specific version of the signature package on Cisco.com (e.g. S293).

	Command or Action	Purpose
		(Optional) The **username** keyword and *name* argument and **password** keyword and *password* argument is for the automatic signature update function.
Step 8	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	Verifies that Cisco IOS IPS is properly configured.
Step 9	**show ip ips auto-update** **Example:** `Router# show ip ips auto-update`	Verifies the signature update configuration.

Configuring Signature Auto Updates to be Upgraded Automatically from Cisco.com

Note Cisco IOS unexpectedly halts all processes or services when signature update S639 or greater is applied. To avoid such an occurrence, Cisco IOS IPS signature auto updates from the Software Download Center is disabled.

You can download the latest IOS IPS Signature updates here: http://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=281442967&catid=268438162&softwareid=280775022

The steps in this task create a new trustpoint (Certificate Authority (CA) server) for a certificate. Repeat the steps in this task to create additional trustpoint certificates. Use meaningful names to differentiate trustpoints. See the examples at the end of this task.

Before You Begin

Ensure that the following pre-requisites are satisfied before configuring the router for upgrading IPS signatures automatically from Cisco.com:

- Ensure that the router points to a reliable time source using Network Time Protocol (NTP). See the Cisco IOS Basic System Management Configuration Guide for more information.
- Ensure that DNS is enabled on the router to resolve domain names to IP addresses. See the Cisco IOS IP Addressing: DNS Configuration Guide for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ida-client server url** *url*
4. **password encryption aes**
5. **key config-key password-encrypt**
6. **ip ips auto-update**
7. **occur-at** {[**monthly** | **weekly**] *days minutes hours*}
8. **cisco**
9. **username** *name* **password** *password*
10. **exit**
11. **show ip ips configuration**
12. **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.
Step 3	**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 4	**password encryption aes** **Example:** `Router(config)# password encryption aes`	Enables the encryption of the password with a type 6 encrypted preshared key.
Step 5	**key config-key password-encrypt** **Example:** `Router(config)# key config-key password-encrypt`	Configures the encrypted preshared key that is used to encrypt all other keys in the router.

	Command or Action	Purpose
Step 6	**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 7	**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 8	**cisco** **Example:** `Router(config-ips-auto-update)# cisco`	Enables automatic Cisco IOS IPS signature updates from Cisco.com.
Step 9	**username** *name* **password** *password* **Example:** `Router(config-ips-auto-update)# username myips password secret`	(Optional) Defines a username and password for the signature update function.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	Verifies that Cisco IOS IPS is properly configured.
Step 12	**show ip ips auto-update** **Example:** `Router# show ip ips auto-update`	Verifies the signature update configuration.

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.

Before You Begin

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.

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

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#
```

Configuring IPS Signatures to be Upgraded Automatically from Cisco.com: Example

The following example shows the part of the running configuration of a router that is configured to have IPS signatures automatically upgraded from Cisco.com. In this example, the distinct CA trustpoints "root" and "sub-root" are configured.

```
ida-client server url https://www.cisco.com/cgi-bin/front.x/ida/locator/locator.pl
ip domain name yourdomain.com
ip name-server 192.168.200.10
ip ips config location flash:/ips retries 1 timeout 1
ip ips notify SDEE
ip ips name IPS
!
ip ips signature-category
  category all
   retired true
  category ios_ips basic
   retired false
!
password encryption aes
!
crypto pki trustpoint root
 enrollment terminal
 revocation-check none
!
crypto pki trustpoint sub-root
 enrollment terminal
 revocation-check none
!
crypto pki certificate chain root
 certificate ca 18DAD19E267DE8BB4A2158CDCC6B3B4A
  308204D3 308203BB A0030201 02021018 DAD19E26 7DE8BB4A 2158CDCC 6B3B4A30
  0D06092A 864886F7 0D010105 05003081 CA310B30 09060355 04061302 55533117
  30150603 55040A13 0E566572 69536967 6E2C2049 6E632E31 1F301D06 0355040B
  13165665 72695369 676E2054 72757374 204E6574 776F726B 313A3038 06035504
  0B133128 63292032 30303620 56657269 5369676E 2C20496E 632E202D 20466F72
  20617574 686F7269 7A656420 75736520 6F6E6C79 31453043 06035504 03133C56
  65726953 69676E20 436C6173 73203320 5075626C 69632050 72696D61 72792043
  65727469 66696361 74696F6E 20417574 686F7269 7479202D 20473530 1E170D30
  36313130 38303030 3030305A 170D3336 30373136 32333539 35395A30 81CA310B
```

```
  30090603 55040613 02555331 17301506 0355040A 130E5665 72695369 676E2C20
  496E632E 311F301D 06035504 0B131656 65726953 69676E20 54727573 74204E65
  74776F72 6B313A30 38060355 040B1331 28632920 32303036 20566572 69536967
  6E2C2049 6E632E20 2D20466F 72206175 74686F72 697A6564 20757365 206F6E6C
  79314530 43060355 0403133C 56657269 5369676E 20436C61 73732033 20507562
  6C696320 5072696D 61727920 43657274 69666963 6174696F 6E204175 74686F72
  69747920 2D204735 30820122 300D0609 2A864886 F70D0101 01050003 82010F00
  3082010A 02820101 00AF2408 08297A35 9E600CAA E74B3B4E DC7CBC3C 451CBB2B
  E0FE2902 F95708A3 64851527 F5F1ADC8 31895D22 E82AAAA6 42B38FF8 B955B7B1
  B74BB3FE 8F7E0757 ECEF43DB 66621561 CF600DA4 D8DEF8E0 C362083D 5413EB49
  CA595485 26E52B8F 1B9FEBF5 A191C233 49D84363 6A524BD2 8FE87051 4DD18969
  7BC770F6 B3DC1274 DB7B5D4B 56D396BF 1577A1B0 F4A225F2 AF1C9267 18E5F406
  04EF90B9 E400E4DD 3AB519FF 02BAF43C EEE08BEB 378BECF4 D7ACF2F6 F03DAFDD
  75913319 1D1C40CB 74241921 93D914FE AC2A52C7 8FD50449 E48D6347 883C6983
  CBFE47BD 2B7E4FC5 95AE0E9D D4D143C0 6773E314 087EE53F 9F73B833 0ACF5D3F
  3487968A EE53E825 15020301 0001A381 B23081AF 300F0603 551D1301 01FF0405
  30030101 FF300E06 03551D0F 0101FF04 04030201 06306D06 082B0601 05050701
  0C046130 5FA15DA0 5B305930 57305516 09696D61 67652F67 69663021 301F3007
  06052B0E 03021A04 148FE5D3 1A86AC8D 8E6BC3CF 806AD448 182C7B19 2E302516
  23687474 703A2F2F 6C6F676F 2E766572 69736967 6E2E636F 6D2F7673 6C6F676F
  2E676966 301D0603 551D0E04 1604147F D365A7C2 DDECBBF0 3009F343 39FA02AF
  33313330 0D06092A 864886F7 0D010105 05000382 01010093 244A305F 62CFD81A
  982F3DEA DC992DBD 77F6A579 2238ECC4 A7A07812 AD620E45 7064C5E7 97662D98
  097E5FAF D6CC2865 F201AA08 1A47DEF9 F97C925A 0869200D D93E6D6E 3C0D6ED8
  E6069140 18B9F8C1 EDDFDB41 AAE09620 C9CD6415 3881C994 EEA28429 0B136F8E
  DB0CDD25 02DBA48B 1944D241 7A05694A 584F60CA 7E826A0B 02AA2517 39B5DB7F
  E784652A 958ABD86 DE5E8116 832D10CC DEFDA882 2A6D281F 0D0BC4E5 E71A2619
  E1F4116F 10B595FC E7420532 DBCE9D51 5E28B69E 85D35BEF A57D4540 728EB70E
  6B0E06FB 33354871 B89D278B C4655F0D 86769C44 7AF6955C F65D3208 33A454B6
  183F685C F2424A85 3854835F D1E82CF2 AC11D6A8 ED636A
        quit
crypto pki certificate chain sub-root
 certificate ca 6ECC7AA5A7032009B8CEBCF4E952D491
  308205EC 308204D4 A0030201 0202106E CC7AA5A7 032009B8 CEBCF4E9 52D49130
  0D06092A 864886F7 0D010105 05003081 CA310B30 09060355 04061302 55533117
  30150603 55040A13 0E566572 69536967 6E2C2049 6E632E31 1F301D06 0355040B
  13165665 72695369 676E2054 72757374 204E6574 776F726B 313A3038 06035504
  0B133128 63292032 30303620 56657269 5369676E 2C20496E 632E202D 20466F72
  20617574 686F7269 7A656420 75736520 6F6E6C79 31453043 06035504 03133C56
  65726953 69676E20 436C6173 73203320 5075626C 69632050 72696D61 72792043
  65727469 66696361 74696F6E 20417574 686F7269 7479202D 20473530 1E170D31
  30303230 38303030 3030305A 170D3230 30323037 32333539 35395A30 81B5310B
  30090603 55040613 02555331 17301506 0355040A 130E5665 72695369 676E2C20
  496E632E 311F301D 06035504 0B131656 65726953 69676E20 54727573 74204E65
  74776F72 6B313B30 39060355 040B1332 5465726D 73206F66 20757365 20617420
  68747470 733A2F2F 7777772E 76657269 7369676E 2E636F6D 2F727061 20286329
  3130312F 302D0603 55040313 26566572 69536967 6E20436C 61737320 33205365
  63757265 20536572 76657220 4341202D 20473330 82012230 0D06092A 864886F7
  0D010101 05000382 010F0030 82010A02 82010100 B187841F C20C45F5 BCAB2597
  A7ADA23E 9CBAF6C1 39B88BCA C2AC56C6 E5BB658E 444F4DCE 6FED094A D4AF4E10
  9C688B2E 957B899B 13CAE234 34C1F35B F3497B62 83488174 D188786C 0253F9BC
  7F432657 5833833B 330A17B0 D04E9124 AD867D64 12DC744A 34A11D0A EA961D0B
  15FCA34B 3BCE6388 D0F82D0C 948610CA B69A3DCA EB379C00 48358629 5078E845
  63CD1941 4FF595EC 7B98D4C4 71B350BE 28B38FA0 B9539CF5 CA2C23A9 FD1406E8
  18B49AE8 3C6E81FD E4CD3536 B351D369 EC12BA56 6E6F9B57 C58B14E7 0EC79CED
  4A546AC9 4DC5BF11 B1AE1C67 81CB4455 33997F24 9B3F5345 7F861AF3 3CFA6D7F
  81F5B84A D3F58537 1CB5A6D0 09E4187B 384EFA0F 02030100 01A38201 DF308201
  DB303406 082B0601 05050701 01042830 26302406 082B0601 05050730 01861868
  7474703A 2F2F6F63 73702E76 65726973 69676E2E 636F6D30 12060355 1D130101
  FF040830 060101FF 02010030 70060355 1D200469 30673065 060B6086 480186F8
  45010717 03305630 2806082B 06010505 07020116 1C687474 70733A2F 2F777777
  2E766572 69736967 6E2E636F 6D2F6370 73302A06 082B0601 05050702 02301E1A
  1C687474 70733A2F 2F777777 2E766572 69736967 6E2E636F 6D2F7270 61303406
  03551D1F 042D302B 3029A027 A0258623 68747470 3A2F2F63 726C2E76 65726973
  69676E2E 636F6D2F 70636133 2D67352E 63726C30 0E060355 1D0F0101 FF040403
  02010630 6D06082B 06010505 07010C04 61305FA1 5DA05B30 59305730 55160969
  6D616765 2F676966 3021301F 30070605 2B0E0302 1A04148F E5D31A86 AC8D8E6B
  C3CF806A D448182C 7B192E30 25162368 7474703A 2F2F6C6F 676F2E76 65726973
  69676E2E 636F6D2F 76736C6F 676F2E67 69663028 0603551D 11042130 1FA41D30
  1B311930 17060355 04031310 56657269 5369676E 4D504B49 2D322D36 301D0603
  551D0E04 1604140D 445C1653 44C1827E 1D20AB25 F40163D8 BE79A530 1F060355
  1D230418 30168014 7FD365A7 C2DDECBB F03009F3 4339FA02 AF333133 300D0609
  2A864886 F70D0101 05050003 82010100 0C8324EF DDC30CD9 589CFE36 B6EB8A80
```

```
  4BD1A3F7 9DF3CC53 EF829EA3 A1E697C1 589D756C E01D1B4C FAD1C12D 05C0EA6E
  B2227055 D9203340 3307C265 83FA8F43 379BEA0E 9A6C70EE F69C803B D937F47A
  6DECD018 7D494ACA 99C71928 A2BED877 24F78526 866D8705 404167D1 273AEDDC
  481D22CD 0B0B8BBC F4B17BFD B499A8E9 762AE11A 2D876E74 D388DD1E 22C6DF16
  B62B8214 0A945CF2 50ECAFCE FF62370D AD65D306 4153ED02 14C8B558 28A1ACE0
  5BECB37F 954AFB03 C8AD26DB E6667812 4AD99F42 FBE198E6 42839B8F 8F6724E8
  6119B5DD CDB50B26 058EC36E C4C875B8 46CFE218 065EA9AE A8819A47 16DE0C28
  6C2527B9 DEB78458 C61F381E A4C4CB66
        quit
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

ip ips auto-update
 occur-at weekly 0-6 10 0-23
 cisco
 username ccouserid password ccouserpw
!
interface GigabitEthernet0/0
 ip address 192.168.200.1 255.255.255.0
 ip ips IPS in
 duplex auto
 speed auto
```

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

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

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

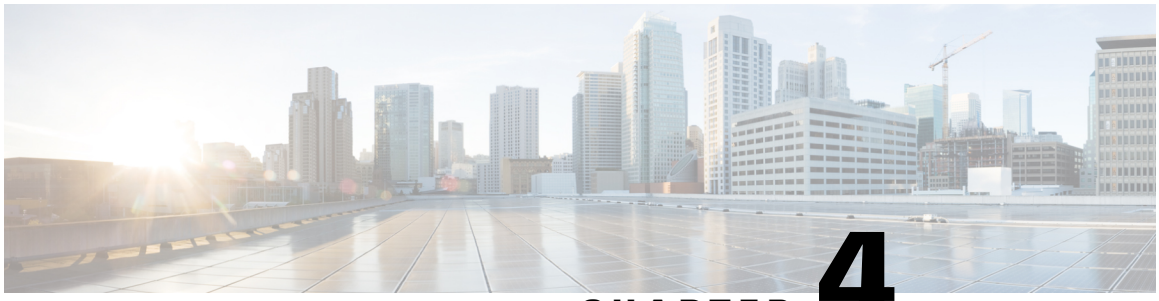
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 7: 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)**.
Direct Download from CCO capability in IOS IPS	15.1(1)T	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 commands were introduced or modifief: **ip ips enable-clidelta**, **show ip ips sig-clidelta**.

Feature Name	Releases	Feature Information
Cisco IOS IPS with Lightweight Signatures	15.2(1)T and 15.2(2)T	This feature was deprecated in Cisco IOS Release 15.2(1)T and 15.2(2)T. The following commands were deprecatef: **ip ips inherit-obsolete tunings**, **ip ips memory regex chaining**, **ip ips memory threshold**.



CHAPTER 4

Cisco IOS IPS Support for Microsoft Engines

The Cisco IOS IPS Support for Microsoft Engines feature extends Cisco IOS Intrusion Prevention Systems (IPS) to support Microsoft RPC (Remote Procedure Call) and Microsoft SMB (Server Message Block) protocols. IPS signatures can now scan for, detect, and take proper action against vulnerabilities in MSRPC and SMB protocols.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 table.

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

Information About Cisco IOS IPS Support for Microsoft Engines

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 via 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.

How to Use Cisco IOS IPS

The addition of the MSRPC and MSB protocol support does not change the way in which Cisco IOS IPS is defined and enabled in your network. For information on how to enable IPS on your network via command-line interface (CLI), see the section "How to Use Cisco IOS 5.x Format Signatures with Cisco IOS IPS" within the document *Cisco IOS IPS 5.x Signature Format Support and Usability Enhancement*.

Configuration Examples for Cisco IOS IPS

show ip ips signature Output to Verify MS Engines Example

The following sample output from the **show ip ips signature** command displays output for the service-msrpc and service-smb-advanced signatures:

```
Signature Micro-Engine: service-msrpc: Total Signatures 21
service-msrpc enabled signatures: 21
service-msrpc compiled signatures: 21
SigIf:SubID En Cmp Action Sev Trait EC AI GST SI SM SW SFR Rel
----------- -- ---- ------ --- ----- ---- ---- ----- --- -- -- --- ---
3330:0 Y Y A HIGH 0 1 0 0 0 FA N 100 S148
3332:0 Y Y A HIGH 0 35 0 0 0 FA N 100 S148
3337:0 Y Y A HIGH 0 8 2 0 0 FA N 100 S85
3331:2 Y Y A HIGH 0 1 0 0 0 FA N 90 S215
3327:12 Y Y A HIGH 0 1 0 0 0 FA N 85 S214
3328:3 Y Y A MED 0 1 0 0 0 FA N 85 S170
3328:1 Y Y A MED 0 1 0 0 0 FA N 85 S148
3327:8 Y Y A INFO 0 1 0 0 0 FA N 85 S214
3334:6 Y Y A HIGH 0 1 0 0 0 FA N 80 S215
3327:0 Y Y A HIGH 0 1 0 0 0 FA N 80 S165
6232:0 Y Y A HIGH 0 1 0 0 0 FA N 75 S209
3327:4 Y Y A HIGH 0 1 0 0 0 FA N 75 S188
3334:5 Y Y A HIGH 0 2 2 0 0 FA N 75 S179
3338:2 Y Y A HIGH 0 40 3 0 0 FA N 75 S175
```

```
3338:3 Y Y A HIGH 0 1 0 0 0 FA N 75 S175
6130:0 Y Y A HIGH 0 1 0 0 0 FA N 75 S167
6130:6 Y Y A INFO 0 1 0 0 0 FA N 75 S192
5567:1 Y Y A INFO 0 1 0 0 0 FA N 55 S187
5567:2 Y Y A INFO 0 1 0 0 0 FA N 55 S187
5567:3 Y Y A INFO 0 1 0 0 0 FA N 55 S187
5567:4 Y Y A INFO 0 1 0 0 0 FA N 55 S187
Signature Micro-Engine: service-smb-advancef: Total Signatures 31
service-smb-advanced enabled signatures: 31
service-smb-advanced compiled signatures: 31
SigIf:SubID En Cmp Action Sev Trait EC AI GST SI SM SW SFR Rel
----------- -- ---- ------ --- ----- ---- ---- ----- --- -- -- --- ---
5593:0 Y Y A HIGH 0 1 0 0 0 FA N 100 S262
5592:0 Y Y A HIGH 0 1 0 0 0 FA N 100 S262
5582:0 Y Y A HIGH 0 1 0 0 0 FA N 100 S262
5599:0 Y Y A HIGH 0 1 0 0 0 FA N 100 S262
5595:0 Y Y A MED 0 1 0 0 0 FA N 100 S262
5579:0 Y Y A INFO 0 1 0 0 0 FA N 100 S264
5581:0 Y Y A INFO 0 1 0 0 0 FA N 100 S264
5580:0 Y Y A INFO 0 1 0 0 0 FA N 100 S264
5584:0 Y Y A INFO 0 1 0 0 0 FA N 100 S262
5576:0 Y Y A INFO 0 1 0 0 0 FA N 100 S262
5577:0 Y Y A INFO 0 1 0 0 0 FA N 100 S262
5583:0 Y Y A INFO 0 1 0 0 0 FA N 100 S262
5591:0 Y Y A INFO 0 1 0 0 0 FA N 100 S262
5590:0 Y Y A INFO 0 1 0 0 0 FA N 100 S262
5598:0 Y Y A HIGH 0 1 0 0 0 FA N 85 S264
5588:0 Y Y A HIGH 0 1 0 0 0 FA N 85 S262
5586:0 Y Y A HIGH 0 1 0 0 0 FA N 85 S262
5585:0 Y Y A MED 0 1 0 0 0 FA N 85 S264
5579:1 Y Y A MED 0 1 0 0 0 FA N 85 S264
5602:0 Y Y A MED 0 1 0 0 0 FA N 85 S262
5589:0 Y Y A LOW 0 1 0 0 0 FA N 85 S262
5578:0 Y Y A INFO 0 1 0 0 0 FA N 85 S264
5605:0 Y Y A INFO 0 1 0 0 0 FA N 85 S262
5600:0 Y Y A HIGH 0 1 0 0 0 FA N 75 S262
5597:0 Y Y A HIGH 0 50 0 0 0 FA N 75 S262
5594:0 Y Y A HIGH 0 1 0 0 0 FA N 75 S262
5587:0 Y Y A HIGH 0 1 0 0 0 FA N 75 S262
5603:0 Y Y A MED 0 1 0 0 0 FA N 75 S262
5591:1 Y Y A INFO 0 1 0 0 0 FA N 75 S262
5575:0 Y Y A INFO 0 1 0 0 0 FA N 75 S262
5590:1 Y Y A INFO 0 1 0 0 0 FA N 75 S262
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

MIBs

MIB	MIBs Link
None	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

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

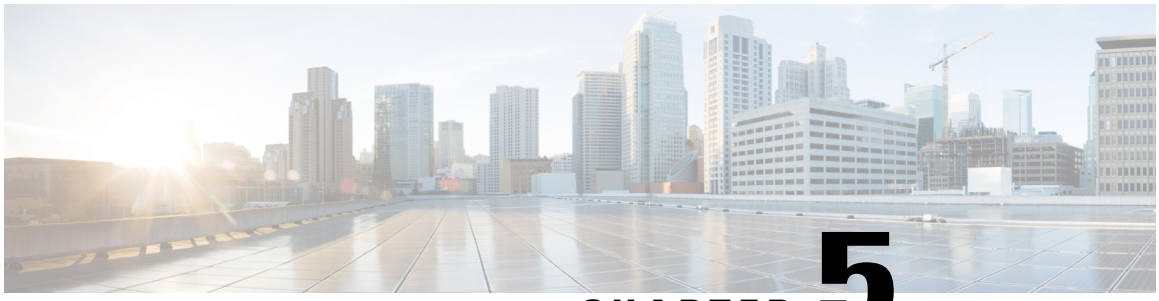
Feature Information for Cisco IOS IPS Support for Microsoft Engines

The following table provides release information about the feature or features described in this module. This table 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.

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

Table 8: Feature Information for Cisco IOS IPS Support for Microsoft Engines

Feature Name	Releases	Feature Information
Cisco IOS IPS Support for Microsoft Engines	12.4(15)T	The Cisco IOS IPS Support for Microsoft Engines feature extends Cisco IPS to support MSRPC and SMB protocols. The following commands were introduced or modifief: **debug ip ips**



CHAPTER 5

VRF Aware Cisco IOS IPS

Virtual Route Forwarding or Virtual Private Network (VPN) Route Forwarding (VRF), is a mechanism that allows multiple instances of a routing table to exist on a router and work simultaneously. This mechanism allows for network paths to be segregated without using multiple devices, thereby increasing network security and eliminating the need for encryption and authentication. VRFs are generally used to create separate VPNs. Allowing Intrusion Prevention System (IPS) to be configured on a per-VRF basis means global parameters will be shared by multiple VPNs, providing VRF related information on the Security Device Event Exchange (SDEE) and syslog alerts.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 table.

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

Prerequisites for VRF Aware Cisco IOS IPS

- Understand Cisco IOS IPS.
- Configure VRFs.
- Verify that the VRFs are operational.
- Verify IPS is supported.
- Capability to send SDEE alarms and syslog with VRF information.
- If two VPN networks have overlapping addresses, VRF-aware network address translation (NAT) is required for them to support VRF Aware Cisco IOS IPS.
- Every VRF instance requires a new IPS rule.

Restrictions for VRF Aware Cisco IOS IPS

- VRF Aware Cisco IOS IPS is not supported on Multiprotocol Label Switching (MPLS) interfaces.

Information About VRF Aware Cisco IOS IPS

Cisco IOS IPS

The Cisco IOS IPS acts as an in-line intrusion detection 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 Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or SDEE. The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. 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 IOS software-based intrusion-prevention capabilities and Cisco IOS firewall are developed 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.

VRF

Each VPN has its own routing and forwarding table in the router, so any customer or site that belongs to a VPN is provided access only to the set of routes contained within that table. Any Provider Edge (PE) router in the MPLS VPN network therefore contains a number of per-VPN routing tables and a global routing table that is used to reach other routers in the service provider network. Effectively, virtual routers are created in a single physical router.

VRF is a Cisco IOS route table instance for connecting a set of sites to a VPN service. A VRF contains a template of a VPN Routing/Forwarding table in a PE router.

The overlapping addresses, usually resulting from the use of private IP addresses in customer networks, are one of the major obstacles to successful deployment of peer-to-peer VPN implementation. The MPLS VPN technology provides a solution to this dilemma.

VRF Lite

VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. VRF Lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be physical, such as Ethernet ports, or logical, such as VLAN switched virtual interfaces (SVIs). However, a Layer 3 interface cannot belong to more than one VRF at a time.

Note VRF Lite interfaces must be Layer 3 interfaces.

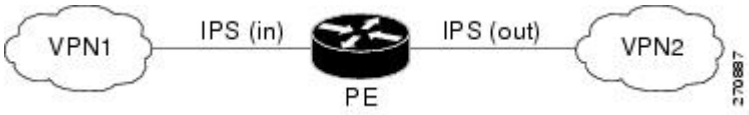
VRF-lite includes these devices:

- Customer edge (CE) devices provide customer access to the service provider network over a data link to one or more PE routers. The CE device advertises the site's local routes to the PE router and learns the remote VPN routes from it. A Catalyst 4500 switch can be a CE device.
- PE routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.
- Provider routers (or core routers) are any routers in the service provider network that do not attach to CE devices.
- The PE is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE to maintain all of the service provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CEs, a PE router exchanges VPN routing information with other PE routers by using Internal BGP (IBPG).

With VRF Lite, multiple customers can share one CE, and only one physical link is used between the CE and the PE. The shared CE maintains separate VRF tables for each customer and switches or routes packets for each customer based on its own routing table. VRF Lite extends limited PE functionality to a CE device, giving the CE device the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

If VRF is configured on an interface and then IPS is attached, an IPS control block is created as shown in the figure below, with an appropriate name for the VRF instead of the existing default value.VRF support for IPS allows customization of IPS parameters, settings and statistics per VRF interface and customization of IPS signature sets for each VRF user. The IPS VRF code takes care of the VRF support including overlapping addresses.

Figure 1: IPS in a VRF-to-VRF Scenario



Applying IPS Directly to a VRF

Virtual Route Forwarding (VRF) is a mechanism that allows multiple instances of a routing table to exist on a router and work simultaneously. This mechanism allows for network paths to be segregated without using multiple devices which increases network security and eliminates the need for encryption and authentication. VRFs are generally used to create separate Virtual Private Networks (VPNs). If VRF is configured on an interface and IPS is attached, an IPS control block is created with the appropriate name for the VRF instead of the existing default value. VRF support for IPS allows customization of IPS parameters, settings and statistics per VRF interface and customization of IPS signature sets for each VRF user. All interfaces share the same global parameters for IPS, but alarms and event log information on the SDEE and syslog alerts carry respective VRF information.

How to VRF Aware Cisco IOS IPS

Configuring a VRF and Applying IPS Directly to the VRF

The following steps are used to configure VRF routing and forwarding tables, configuring VRF on an interface and attach IPS to this interface:

Note If a global VRF is removed, the IPS configuration on the interfaces belonging to that VRF is cleaned and removed, including any sessions and statistics created on the VRF. The user must reconfigure IPS on the affected interfaces if they are to be used again.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf vrfname**
4. **rd route-distinguisher**
5. **route-target export target VPN extended community**
6. **route-target import target VPN extended community**
7. **exit**
8. **interface FastEthernet port**
9. **ip vrf forwarding vrfname**
10. **ip address range**
11. **ip ips** *ips-name* **in**
12. **exit**
13. **end**

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 vrf vrfname** **Example:** `Router# ip vrf VRF600`	Configures a VRF table and enters VRF configuration mode.
Step 4	**rd route-distinguisher** **Example:** `Router# rd 100:600`	Creates routing and forwarding tables for the VRF instance. **Note** There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).

	Command or Action	Purpose
Step 5	**route-target export target VPN extended community** **Example:** `Router(config-vrf)# route-target export 100:600`	Creates lists of export route-target extended communities for the specified VRF.
Step 6	**route-target import target VPN extended community** **Example:** `Router(config-vrf)#route-target import 100:600`	Creates lists of import route-target extended communities for the specified VRF.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode.
Step 8	**interface FastEthernet port** **Example:** `Router(config)# interface FastEthernet0/1.600`	Enters subinterface configuration mode and specifies a subinterface that is associated with a VRF.
Step 9	**ip vrf forwarding vrfname** **Example:** `Router(config-subif)# ip vrf forwarding VRF600`	Configures the forwarding details for the respective interfaces.
Step 10	**ip address range** **Example:** `Router(config-subif)# ip address 192.168.25.3 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 11	**ip ips** *ips-name* **in** **Example:** `Router(config-subif)# ip ips ips_policy600 in`	Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines.

	Command or Action	Purpose

Step 12	**exit**
	
	Example:
	`Router(config-subif)# exit`
	
	Example:
	`Router(config-if)# end`
		Exits subinterface mode, and enters interface configuration mode.

Step 13	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode.

Configuration Examples for VRF Aware Cisco IOS IPS

Example Cisco IOS IPS Configuration

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_STARTEf: 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
       Enablef: 0
       Compilef: 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_STARTEf: 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_UNSUPPORTEf: 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
       Enablef: 3
       Retiref: 3
Signature Micro-Engine: service-http
     Total Signatures: 611
       Enablef: 159
       Retiref: 428
       Compilef: 183
Signature Micro-Engine: string-tcp
     Total Signatures: 864
       Enablef: 414
       Retiref: 753
       Compilef: 111
Signature Micro-Engine: string-udp
     Total Signatures: 74
       Enablef: 1
       Retiref: 44
       Compilef: 30
Signature Micro-Engine: state
     Total Signatures: 28
       Enablef: 16
       Retiref: 25
       Compilef: 3
Signature Micro-Engine: atomic-ip
     Total Signatures: 252
       Enablef: 56
       Retiref: 148
       Compilef: 103
       Inactive - invalid params: 1
Signature Micro-Engine: string-icmp
     Total Signatures: 3
       Enablef: 0
       Retiref: 2
       Compilef: 1
Signature Micro-Engine: service-ftp
     Total Signatures: 3
       Enablef: 1
       Compilef: 3
Signature Micro-Engine: service-rpc
            Total Signatures: 75
       Enablef: 44
       Retiref: 44
       Compilef: 31
Signature Micro-Engine: service-dns
     Total Signatures: 38
       Enablef: 30
```

```
       Retiref: 5
       Compilef: 33
Signature Micro-Engine: normalizer
     Total Signatures: 9
       Enablef: 8
       Retiref: 5
       Compilef: 4
Signature Micro-Engine: service-smb-advanced (INACTIVE)
Signature Micro-Engine: service-msrpc
     Total Signatures: 22
       Enablef: 22
       Retiref: 22
```

Example VRF Aware Cisco IOS IPS Configuration Without Subinterfaces

The following example shows a physical interface supporting a VRF forwarding table with IPS enabled on VRF600:

```
Router(config)# ip vrf
 VRF600
Router(config-vrf)# rd
 100:600
Router(config-vrf)# route-target export
 100:600
Router(config-vrf)# route-target import
100:600
Router(config-vrf)# exit
Router(config)# interface FastEthernet
0/1
Router(config-subif)# ip vrf forwarding
 VRF600
Router(config-subif)# ip address
 192.168.00.3 192.168.255.225
Router(config-subif)# ip ips
 ips_policy600 in

router(config-subif)# exit

router(config-if)# end
```

Example VRF Aware Cisco IOS IPS Configuration with Subinterfaces

The following example shows two physical interfaces supporting two VRF forwarding tables, VRF600 and VRF601, with IPS enabled on only VRF600:

```
config terminal
Router1(config)# ip vrf
 VRF600
Router1(config-vrf)# rd
 100:600
Router1(config-vrf)# route-target export
 100:600
Router1(config-vrf)# route-target import
 100:600
Router1(config-vrf)# exit
Router1(config)# ip vrf
VRF601
Router1(config-vrf)# rd
 100:601
Router1(config-vrf)# route-target export
100:601
Router1(config-vrf)# route-target import
```

```
100:601
Router1(config-vrf)# exit
Router1(config)# interface FastEthernet
0/0.600
Router1(config-subif)# encapsulation dot1Q
600
Router1(config-subif)# ip vrf forwarding
VRF600
Router1(config-subif)# ip address
 192.168.00.0 192.168.255.0
Router1(config-subif)# exit
Router1(config)# interface FastEthernet
0/0.601
Router1(config-subif)# encapsulation dot1Q
601
Router1(config-subif)# ip vrf forwarding
 VRF601
Rrouter1(config-subif)# ip address
192.168.00.0 192.168.255.0
Router1(config-subif)# end


config terminal

Router2(config)# ip ips name
 ips_policy600

Router2(config)# ip vrf
 VRF600
Router2(config-vrf)# rd
 100:600
Router2(config-vrf)# route-target export
 100:600
Router2(config-vrf)# route-target import
 100:600
Router2(config-vrf)# exit

Router2(config)# ip vrf
VRF601
Router2(config-vrf)# rd
 100:601
Router2(config-vrf)# route-target export
 100:601
Router2(config-vrf)# route-target import
 100:601
Router2(config-vrf)# exit

Router2(config)# interface FastEthernet
0/1.600
Router2(config-subif)# encapsulation dot1Q
 600
Router2(config-subif)# ip vrf forwarding
 VRF600
Router2(config-subif)# ip address
192.168.00.0 192.168.255.0
Router2(config-subif)# ip ips
ips_policy600 in
Router2(config-subif)# exit

Router2(config)# interface FastEthernet
0/1.601
Router2(config-subif)# encapsulation dot1Q
 601
Router2(config-subif)# ip vrf forwarding
 VRF601
Router2(config-subif)# ip address
 192.168.00.0 192.168.255.0
Router2(config-subif)# end
```

Example Multi VRF with IPS and Zone Based Policy (ZBP) Firewall

The following example shows Multiple VRFs configured with IPS and ZBP firewalls:

```
ip cef
!
ip vrf VRF_600
 rd 100:110
 route-target export 100:1000
 route-target import 100:1000
!
ip vrf VRF_601
 rd 100:120
 route-target export 100:2000
 route-target import 100:2000
!
ip ips config location flash:ips5/ retries 1
ip ips name IPS_POLICY_201
ip ips name IPS_POLICY_VRF_600
ip ips name IPS_POLICY_VRF_601
!
ip ips signature-category
  category all
   retired true
!
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
!
class-map type inspect match-any L4-cmti
 match protocol tcp
 match protocol udp
 match protocol icmp
!
policy-map type inspect inside201-outside-pmti
 class type inspect L4-cmti
  inspect
!
policy-map type inspect inside600-outside-pmti
 class type inspect L4-cmti
  inspect
!
policy-map type inspect inside602-outside-pmti
 class type inspect L4-cmti
  inspect
!
zone security inside201
zone security inside600
zone security inside601
zone security outside
!
zone-pair security inside201-outside source inside201 destination outside
 service-policy type inspect inside201-outside-pmti
!
zone-pair security inside600-outside source inside600 destination outside
 service-policy type inspect inside600-outside-pmti
!
zone-pair security inside601-outside source inside602 destination outside
```

```
 service-policy type inspect inside602-outside-pmti
!
interface Loopback0
 ip address 10.10.10.4 10.255.255.255
 ip router isis
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 media-type rj45
 no keepalive
!
interface GigabitEthernet0/0.201
 encapsulation dot1Q 201
 ip address 192.168.00.0 192.168.255.0
 zone-member security inside201
 ip ips myips201 in
 ip ips myips201 out
!
interface GigabitEthernet0/0.600
 encapsulation dot1Q 600
 ip vrf forwarding VRF_600
 ip address 10.0.0.0 10.255.255.255
 zone-member security inside600
 ip ips IPS_POLICY_VRF_600 in
 ip ips IPS_POLICY_VRF_600 out
!
interface GigabitEthernet0/0.601
 encapsulation dot1Q 601
 ip vrf forwarding IPS_POLICY_VRF_601
 ip address 10.0.0.0 10.255.255.255
 zone-member security inside602
 ip ips IPS_POLICY_VRF_601 in
 ip ips IPS_POLICY_VRF_601 out
!
!
interface FastEthernet2/0
 ip address 10.1.1.14 10.255.255.255
 ip router isis
 duplex auto
 speed auto
 mpls ip
!
router isis
 net 10.225.225.225
 is-type level-1
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.10.10.6 remote-as 100
 neighbor 10.10.10.6 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.10.10.6 activate
 neighbor 10.10.10.6 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VRF_600
 redistribute connected
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VRF_601
 redistribute connected
 no synchronization
 exit-address-family
```

Examples VRF Aware Cisco IOS IPS Output and Error Message

Note All VRFs will share the same global IPS configurations, therefore, some show commands which show the information of the global items are shown for every VRF irrespective of whether the event happened on that particular VRF.

Examples VRF Aware Cisco IOS IPS Output

The following is sample output from the **show ip ips statistics**command. The output provides statistics that may not necessarily be the ones that fired on VRF 600.

```
Router# show ip ips statistics vrf VRF_600
Signature statistics [process switch:fast switch]
signature 5170:1 packets checkef: [0:4]
Interfaces configured for ips 4
Session creations since subsystem startup or last reset 0
Current session counts (estab/half-open/terminating) [0:0:0]
Maxever session counts (estab/half-open/terminating) [0:0:0]
Last session created never
Last statistic reset never
TCP reassembly statistics
received 0 packets out-of-order; dropped 0
peak memory usage 0 KB; current usage: 0 KB
peak queue length 0
```

The following is sample output from the **show ip ips interfaces command.**

```
Router# show ip ips interface
Interface Configuration
Interface GigabitEthernet0/0
VRF name: vrf1
Inbound IPS rule is tst
Outgoing IPS rule is not set
```

The following is sample output from the **show ip ips session** command.

```
Router# show ip ips session vrf vrf1
Established Sessions
Session 485EBEE8 (172.16.0.0:10001)=>(172.31.255.255:80) tcp SIS_OPEN
```

The following is sample output from the **clear ip ips statistics**command**.**

```
Router# clear ip ips statistics vrf vrf1
Router# show ip ips stat vrf vrf1

Interfaces configured for ips 1
Session creations since subsystem startup or last reset 0
Current session counts (estab/half-open/terminating) [1:0:0]
Maxever session counts (estab/half-open/terminating) [1:0:0]
Last session created 00:00:26
Last statistic reset 00:00:01
```

Examples ErrMSG with VRF Name Output

The following is sample error message output with the VRF name.

```
%IPS-4-SIGNATURE: Sig:5405 Subsig:0 Sev:100  [192.168.103.1:51129 -> 192.168.3.4:80]
VRF:vrf600 RiskRating:100
```

Examples SDEE Messages with VRF Name

The SDEE messages have been enhanced to show the VRF name. An example of output from the browser is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
  <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
     <env:Body>
         <sf:events xmlns:cid="http://www.cisco.com/cids/2003/08/cidee"
xmlns:sd="http://example.org/2003/08/sdee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://example.org/2003/08/sdee sdee.xsd
http://www.cisco.com/cids/2003/08/cidee cidee.xsd">
              <sf:evIdsAlert eventId="11630069224" vendor="Cisco" severity="unknown">
              <sf:originator>
                  <sf:hostId>iosfw-28a</sf:hostId>
               </sf:originator>
                <sf:time offset="0" timeZone="UTC">1163006922225223338</sf:time>
                 <sf:signature description="" id="5123" version="S4">
                    <cif:subsigId>0</cif:subsigId>
                    <cif:sigDetails>Host:\x3c250+ chars></cif:sigDetails>
                 </sf:signature>
                 <cif:protocol>tcp</cif:protocol>
                 <cif:riskRatingValue>85</cif:riskRatingValue>
                 <sf:participants>
                    <sf:attacker>
       <sf:addr>10.1.0.1</sf:addr>
                               <sf:port>10001</sf:port>
                     </sf:attacker>
                     <sf:target>
                            <sf:addr>10.2.0.1</sf:addr>
                            <sf:port>80</sf:port>
                     </sf:target>
               </sf:participants>
               <sf:actions></sf:actions>
                <cif:interface>Gi0/1</cif:interface>
                <cif:vrf_name>vrf1</cif:vrf_name>
              </sf:evIdsAlert>
      </sf:events>
    </env:Body>
</env:Envelope>
```

Examples SDEE show Commands

The SDEE show commands have been enhanced to include VRF specific information, the following is sample output from the **show ip sdee alerts**commanf:

```
Router# show ip sdee alerts

Alert storage: 200 alerts using 56000 bytes of memory
                                    SDEE Alerts
      SigID    Sig Name                    SrcIP:SrcPort         DstIP:DstPort     VRF
                                                 or Summary Info
   1:  5170:1                              192.162.4.0:3692       192.162.6.0:80    NONE
```

```
   2:  5170:1                                192.162.4.0:3692       192.162.6.0:80 VRF_600
```

The following is sample output from the **show ip sdee events** command.

```
Router# show ip sdee events

Alert storage:   200 alerts using 56000 bytes of memory
Message storage: 200 messages using 84800 bytes of memory
                                          SDEE Events
      Time                          Type    Description
1: 10:25:55 MST Jan 22 2007  ALERT   Sig ID     5170:1
2: 10:25:57 MST Jan 22 2007  ALERT   Sig ID     5170:1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

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 VRF Aware Cisco IOS IPS

The following table provides release information about the feature or features described in this module. This table 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.

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

Table 9: Feature Information for VRF Aware Cisco IOS IPS

Feature Name	Releases	Feature Information
VRF aware Cisco IOS IPS	12.4(20)T	Virtual Route Forwarding or Virtual Private Network (VPN) Route Forwarding (VRF), is a mechanism that allows multiple instances of a routing table to exist on a router and work simultaneously. This mechanism allows for network paths to be segregated without using multiple devices, thereby increasing network security and eliminating the need for encryption and authentication. VRFs are generally used to create separate VPNs. Allowing Intrusion Prevention System (IPS) to be configured on a per-VRF basis means global parameters will be shared by multiple VPNs, providing VRF related information on the Security Device Event Exchange (SDEE) and syslog alerts. The following commands were introduced or modifief: **clear ip ips statistics**, **show ip ips**