Configuring Cisco IOS Intrusion Prevention System

Last Updated: August 14, 2012

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 at the end of this module.

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.

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

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

VMS IDS MC provides a web-based interface for configuring, managing, and monitoring multiple IDS sensors. 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

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 ID: 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 is 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

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 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.

SDEE

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, page 4

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.

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 the transparent IPS does not support Layer 2 IPS functionality; thus, IPS can act only as a Layer 3 device.)

- Transparent Bridging Overview, page 4
- Transparent and Non-Transparent IPS Devices Configured on the Same Router, page 4

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 Router

A transparent IPS device supports a 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 nontransparent IPS device to be configured on the same router. 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 router 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 router 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 which SDF that should be used the first time IPS in enabled on the router.

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

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

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

The **attack-drop.sdf**, **128MB.sdf**, and **256MB.sdf** are available in flash on all Cisco access routers that are shipped with Cisco IOS Release 12.4 or later. 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 router refers to the built-in signatures within the Cisco IOS image.

The SDF file can also be downloaded onto your router 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 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 router.

Signature Microengines Overview and Lists of Supported Engines

Cisco IOS IPS uses SMEs to load the SDF and scan signatures. Signatures contained 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, page 6

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	--
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	--

The table below lists support for the 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.

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.**

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
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.

Signature ID	Count	Signature Engine
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.

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 Supported in Cisco IOS Release 12.3(8)T

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.

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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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 into your system may be in progress.

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 .

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.

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

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.

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

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

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.

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

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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.
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.

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

Signature ID: SubSig ID	Signature Name	Action[3]	SME	Signature Description
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.

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 Router.”

- 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).”

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

- 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”

After you have configured Cisco IOS IPS on your router, see the following optional sections:

- Configuring a Bridge Group for Transparent Cisco IOS IPS, page 25
- Installing Cisco IOS IPS on a New Router, page 28
- Upgrading to the Latest Cisco IOS IPS Signature Definition File (SDF), page 30
- Merging Built-In Signatures with the attack-drop.sdf File, page 32
- Monitoring Cisco IOS IPS Signatures through Syslog Messages or SDEE, page 36
- Troubleshooting Cisco IOS IPS, page 37

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**

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	**bridge** *bridge-group* **protocol** {**dec** | **ibm** | **ieee** | **vlan-bridge** **Example:** `Router(config)# bridge 1 protocol ieee`	Defines the type of Spanning Tree Protocol (STP).
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet0`	Configures an interface type and enters interface configuration mode.
Step 5	**bridge-group** *bridge-group* **Example:** `Router(config-if)# bridge-group 1`	Assigns each network interface to a bridge group. **Note** Complete Step 4 and Step 5 for each interface 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:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**bridge irb** **Example:** `Router(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.

	Command or Action	Purpose
Step 8	**bridge** *bridge-group* **route** *protocol* **Example:** `Router(config)# bridge 1 route ip`	Enables the routing of a specified protocol in a specified bridge group.
Step 9	**interface** *type number* **Example:** `Router(config)# interface BVI1`	Configures a BVI and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# no shutdown`	Restarts a disabled interface.

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.

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


- Troubleshooting Tips, page 27
- What to Do Next, page 28

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 Router

Perform this task to install the latest Cisco IOS IPS signatures on a router 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. **exit**
10. **show ip ips configuration**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip ips sdf location** *url* **Example:** `Router(config)# ip ips sdf location disk2:attack-drop.sdf`	(Optional) Specifies the location in which the router loads the SDF, "attack-drop.sdf." **Note** If this command is not issued, the router loads the default, built-in signatures.
Step 4	**ip ips name ips-name** [**list** *acl*] **Example:** `Router(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:** `Router(config)# ip ips signature 1000 disable`	(Optional) Attaches a policy to a given signature.
Step 6	**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 7	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/1`	Configures an interface type and enters interface configuration mode.
Step 8	**ip ips** *ips-name* {**in** | **out**} **Example:** `Router(config-if)# ip ips MYIPS in`	Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines. **Note** Whenever signatures are replaced or merged, the router prompt is suspended while the signature engines for the newly added or merged signatures are being built. The router prompt is available again after the engines are built. Depending on your platform and how many signatures are being loaded, building the engine can take up to several seconds. It is recommended that you enable logging messages to monitor the engine building status.

	Command or Action	Purpose

Step 9	**exit**
	
	Example:
	`Router(config-if)# exit`
	
	Example:
	`Router(config)# exit`
		Exits interface and global configuration modes.

Step 10	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	(Optional) Verifies that Cisco IOS IPS is properly configured.

Upgrading to the Latest Cisco IOS IPS Signature Definition File (SDF)

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.
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.

	Command or Action	Purpose
Step 7	**ip ips deny-action ips-interface** **Example:** `Router(config)# ip ips deny-action ips-interface`	(Optional) C reates 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.
Step 11	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	(Optional) Verifies that Cisco IOS IPS is properly configured.
Step 12	**show ip ips signatures** [**detailed**] **Example:** `Router# show ip ips signatures`	(Optional) Verifies 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 Router".

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*}
   - **list** *acl* --Packets that are permitted through a specified ACL is scanned by IPS.
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:** `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	**no ip ips location in builtin** **Example:** `Router(config)# no ip ips location in builtin`	(Optional) Instructs the router not to 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.

	Command or Action	Purpose
Step 4	**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 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 6	**copy** [**/erase**] *url* **ips-sdf** **Example:** `Router# copy disk2:attack-drop.sdf ips-sdf`	Loads the SDF in the router. The SDF merges with the signatures that are already loaded in the router, 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 router is reloaded, it refers to a previously specified SDF location in the configuration or it loads the built-in signatures. **Note** Whenever signatures are replaced or merged, the router prompt is suspended while the signature engines for the newly added or merged signatures are being built. The router prompt is available again after the engines are built. Depending on your platform and how many signatures are being loaded, building the engine can take up to several seconds. It is recommended that you enable logging messages to monitor the engine building status.
Step 7	**copy ips-sdf** *url* **Example:** `Router# 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:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 9	**ip ips signature** *signature-id* [**:***sub-signature-id*] {**delete** | **disable** | **list** *acl-list*} • **list** *acl* --Packets that are permitted through a specified ACL is scanned by IPS. **Example:** `Router(config)# ip ips signature 1107 disable`	(Optional) Instructs the router to scan for the specified signature but not take any action if the signature is detected.
Step 10	**ip ips sdf location** *url* **Example:** `Router(config)# ip ips sdf location disk2:my-signatures.sdf`	Configures the router to initialize the new SDF.
Step 11	**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 12	**interface** *type name* **Example:** `Router(config)# interface GigabitEthernet0/1`	Configures an interface type and enters interface configuration mode.
Step 13	**ip ips** *ips-name* {**in** | **out**} **Example:** `Router(config-if)# ip ips MYIPS in`	Applies an IPS rule at an interface and reloads the router and reinitializes Cisco IOS IPS. **Note** The router prompt disappears while the signatures are loading and the signature engines are building. The router prompt reappears after the signatures have been loaded and the signature engines have been built.
Step 14	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode.

	Command or Action	Purpose
Step 15	**show ip ips signatures** [**detailed**] **Example:** `Router# 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.
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.

	Command or Action	Purpose
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, page 37

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, page 37
- Conditions of an SME Build Failure, page 39

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_SKIPPED: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_UNSCANNED: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.
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.

System Message	Description
`%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_FAILED: 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_FAILED: 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_FAILED: 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-DISABLED: 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

Loading the Default Signatures Example

The following example shows the Cisco IOS IPS commands required 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.

```
!
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
!
```

Loading the attack-drop.sdf Example

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
!
```

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

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-DISABLED:IPS removed from all interfaces - IPS disabled
!
Router(config-if)# ip ips MYIPS in
!
Router(config-if)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*

Related Topic	Document Title
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	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**

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