Zone-Based Policy Firewall

First Published: February 22, 2006
Last Updated: March 25, 2011

This module describes the Cisco IOS unidirectional firewall policy between groups of interfaces known as zones. Prior to the release of Cisco IOS unidirectional firewall policy, Cisco IOS firewalls were configured as an inspect rule only on interfaces. Traffic entering or leaving the configured interface was inspected based on the direction that the inspect rule was applied.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Zone-Based Policy Firewall" section on page 65.

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

Contents

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

Prerequisites for Zone-Based Policy Firewall

- Before you create zones, you must consider what should constitute the zones. The general guideline is that you should group interfaces that are similar when they are viewed from a security perspective.
- The Wide Area Application Services (WAAS) and Cisco IOS firewall interoperability capability applies only on the Cisco IOS Zone-Based Policy Firewall feature in Release 12.4(11)T2 and later releases. The Cisco IOS firewall that preceded Release 12.4(11)T2 does not incorporate the Cisco WAAS interoperability enhancement.

Restrictions for Zone-Based Policy Firewall

- If a configuration includes both security zones and inspect rules on interfaces (the old methodology), the configuration may work, but that type of configuration is not recommend.
- The cumulative counters in the **show policy-map type inspect zone-pair** command output do not increment for **match** statements in a nested class-map configuration in Cisco IOS Releases 12.4(20)T and 12.4(15)T. The problem with the counters exists regardless of whether the top level class map uses the **match-any** or **match-all** keyword. Refer to the "Example: Protocol Match Data Not Incrementing for a Class Map" section on page 63 for more information.
- In Cisco IOS Release 12.4(15)T, if Simple Mail Transfer Protocol (SMTP) is configured for inspection in a class map and the inspection of Extended Simple Mail Transfer Protocol (ESMTP) needs to be configured, then the **no match protocol smtp** command must be entered before adding the **match protocol smtp extended** command. To revert to regular SMTP inspection, use the **no match protocol smtp extended** command and then enter the **match protocol smtp** command.

  If these commands are not configured in the proper order in a particular release, then the following error displays:

  ```
  %Cannot add this filter.Remove match protocol smtp filter and then add this filter
  ```
- In a WAAS and Cisco IOS firewall configuration, all packets processed by a Wide Area Application Engine (WAE) device must go over the Cisco IOS firewall in both directions to support the Web Cache Coordination Protocol (WCCP). This situation occurs because the Layer 2 redirect is not available in Release 12.4T. If Layer 2 redirect is configured on the WAE, the system defaults to the generic routing encapsulation (GRE) redirect to continue to function.
- When an in-to-out zone-based policy is configured to match the Internet Control Message Protocol (ICMP) on a Windows system, the **traceroute** command works. However, the same configuration on an Apple system does not work because it uses a UDP-based traceroute. To overcome this issue, configure an out-to-in zone-based policy with the **icmp time-exceeded** and **icmp host unreachable** commands with the **pass** command (not the **inspect** command).
- In a WAAS and Cisco IOS firewall configuration, WCCP does not support traffic redirection using policy-based routing (PBR).
- Stateful inspection support for multicast traffic is not supported between any zones, including the self zone. Use Control Plane Policing for protection of the control plane against multicast traffic.
- A UDP-based traceroute is not supported through ICMP inspection.
- To allow GRE and Encapsulating Security Payload (ESP) protocol traffic through a zone-based policy firewall, you must use the **pass** command. The GRE and ESP protocols do not support stateful inspection. Hence, if you use the **inspect** command, the traffic for these protocols is dropped.

Information About Zone-Based Policy Firewall

Top-Level Class Maps and Policy Maps

Top-level class maps allow you to identify the traffic stream at a high level. This is accomplished by using **match access-group** and **match protocol** commands. Top-level class maps are also referred to as Layer 3 and Layer 4 class maps.

Top-level policy maps allow you to define high-level actions by using the **inspect**, **drop**, **pass**, and **urlfilter** keywords. You can attach the maps to a target (zone pair).

Note Only inspect type policies can be configured on a zone pair.

Application-Specific Class Maps and Policy Maps

Application-specific class maps allow you to identify traffic based on the attributes of a given protocol. All the match conditions in these class maps are specific to an application (for example, HTTP or SMTP). Application-specific class maps are identified by an additional subtype that generally is the protocol name (HTTP or SMTP) in addition to the type **inspect**.

Application-specific policy maps are used to specify a policy for an application protocol. For example, if you want to drop HTTP traffic with Unique Resource Identifier (URI) lengths exceeding 256 bytes, you must configure an HTTP policy map to do that. Application-specific policy maps cannot be attached directly to a target (zone pair). They must be configured as “child” policies in a top-level Layer 3 or Layer 4 policy map.

Overview of Zones

A zone is a group of interfaces that have similar functions or features. Zones provide a way for you to specify where a Cisco IOS firewall is applied.

For example, on a router, Gigabit Ethernet interface 0/0/0 and Gigabit Ethernet interface 0/0/1 may be connected to the local LAN. These two interfaces are similar because they represent the internal network, so they can be grouped into a zone for firewall configurations.

By default, the traffic between interfaces in the same zone is not subjected to any policy. The traffic passes freely. Firewall zones are used for security features.

Note Zones may not span interfaces in different VPN routing and forwarding (VRF) instances.

When a zone-based policy firewall is enabled for TCP keepalive traffic and the host behind the firewall is undergoing an ungraceful disconnect, TCP keepalive works only when the configured TCP timeout is complete. On receiving an out of window reset (RST) packet, the firewall sends an empty acknowledge (ACK) packet to the initiator of the RST packet. This ACK has the current sequence (SEQ) and ACK number from the firewall session. On receiving this ACK, the client sends an RST packet with the SEQ number that is equal to the ACK number in the ACK packet. The firewall processes this RST packet, clears the firewall session, and passes the RST packet.

Security Zones

A security zone is a group of interfaces to which a policy can be applied.

Grouping interfaces into zones involves two procedures:

- Creating a zone so that interfaces can be attached to it
- Configuring an interface to be a member of a given zone

By default, traffic flows among interfaces that are members of the same zone.

When an interface is a member of a security zone, all traffic to and from that interface (except traffic going to the router or initiated by the router) is dropped. To permit traffic to and from a zone-member interface, you must make that zone part of a zone pair and then apply a policy to that zone pair. If the policy permits traffic (through **inspect** or **pass** actions), traffic can flow through the interface.

Basic rules to consider when setting up your zones are as follows:

- Traffic from a zone interface to a nonzone interface or from a nonzone interface to a zone interface is always dropped.
- Traffic between two zone interfaces is inspected if there is a zone pair relationship for each zone and if there is a configured policy for that zone pair.
- By default, all traffic between two interfaces in the same zone is always allowed as if the “pass” action is configured.
- A zone pair can be configured with a zone as both the source and the destination zones. An inspect policy can be configured on this zone pair to inspect or drop the traffic between two interfaces in the same zone.

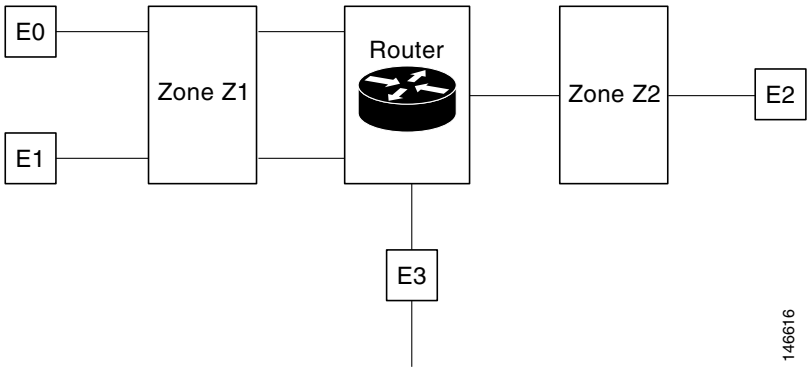
For traffic to flow among all the interfaces in a router, all the interfaces must be a member of one security zone or another.

It is not necessary for all router interfaces to be members of security zones.

Figure 1 illustrates the following:

- Interfaces E0 and E1 are members of security zone Z1.
- Interface E2 is a member of security zone Z2.
- Interface E3 is not a member of any security zone.

Figure 1 Security Zone Restrictions



The following situations exist:

- The zone pair and policy are configured in the same zone. Traffic flows freely between interfaces E0 and E1 because they are members of the same security zone (Z1).
- If no policies are configured, traffic will not flow between any other interfaces (for example, E0 and E2, E1 and E2, E3 and E1, and E3 and E2).
- Traffic can flow between E0 or E1 and E2 only when an explicit policy permitting traffic is configured between zone Z1 and zone Z2.
- Traffic can never flow between E3 and E0/E1/E2 unless default zones are enabled.

Virtual Interfaces as Members of Security Zones

A virtual template interface is a logical interface configured with generic configuration information for a specific purpose or for configuration common to specific users, plus router-dependent information. The template contains Cisco IOS software interface commands that are applied to virtual access interfaces, as needed. To configure a virtual template interface, use the **interface virtual-template** command.

Zone member information is acquired from a RADIUS server and then the dynamically created interface is made a member of that zone.

The **zone-member security** command puts the dynamic interface into the corresponding zone.

Zone Pairs

A zone pair allows you to specify a unidirectional firewall policy between two security zones.

To define a zone pair, use the **zone-pair security** command. The direction of the traffic is specified by specifying a source and destination zone. The source and destination zones of a zone pair must be security zones.

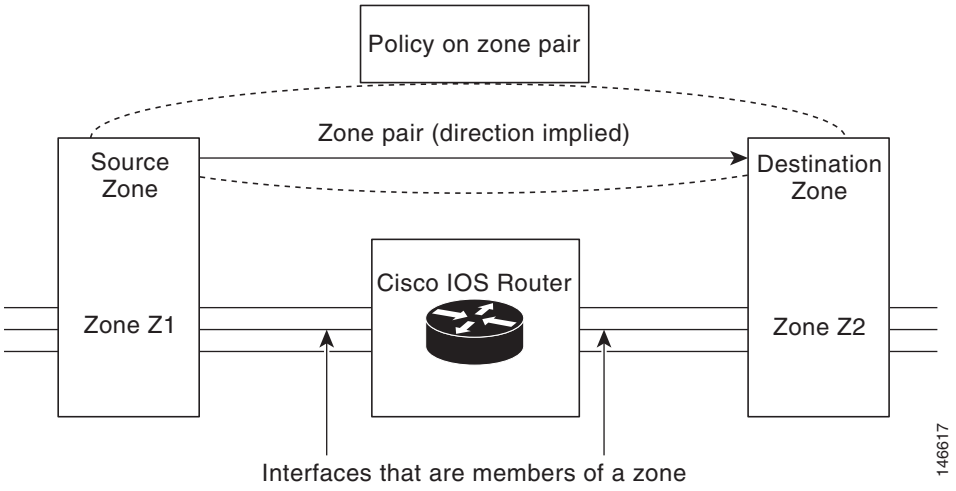
If desired, you can select the default or self zone as either the source or the destination zone. The self zone is a system-defined zone. It does not have any interfaces as members. A zone pair that includes the self zone, along with the associated policy, applies to traffic directed to the router or traffic generated by the router. It does not apply to traffic through the router.

The most common usage of firewalls is to apply them to traffic through a router, so you usually need at least two zones (that is, you cannot use the self zone).

To permit traffic between zone-member interfaces, you must configure a policy permitting (or inspecting) traffic between that zone and another zone. To attach a firewall policy map to the target zone pair, use the **service-policy type inspect** command.

Figure 2 shows the application of a firewall policy to traffic flowing from zone Z1 to zone Z2, which means that the ingress interface for the traffic is a member of zone Z1 and the egress interface is a member of zone Z2.

Figure 2 Zone Pairs



If there are two zones and you require policies for traffic going in both directions (from Z1 to Z2 and Z2 to Z1), you must configure two zone pairs (one for each direction).

If a policy is not configured between a pair of zones, traffic is dropped. However, it is not necessary to configure a zone pair and a service policy solely for return traffic. Return traffic is allowed, by default, if a service policy permits the traffic in the forward direction. In Figure 2, it is not mandatory that you configure a zone pair source Z2 destination Z1 solely for allowing return traffic from Z2 to Z1. The service policy on the Z1-Z2 zone pair takes care of it.

Zones and Inspection

Zone-based policy firewalls examine the source and destination zones from the ingress and egress interfaces for a firewall policy. It is not necessary that all traffic flowing to or from an interface be inspected; you can designate that individual flows in a zone pair be inspected through your policy map that you apply across the zone pair. The policy map will contain class maps that specify the individual flows.

You can also configure **inspect** parameters like TCP thresholds and timeouts on a per-flow basis.

Zones and ACLs

Pinholes are not punched for return traffic in interface access control lists (ACLs).

ACLs applied to interfaces that are members of zones are processed before the policy is applied on the zone pair. So, you must relax interface ACLs when there are policies between zones so that they cannot interfere with the policy firewall traffic.

Zones and VRF-Aware Firewalls

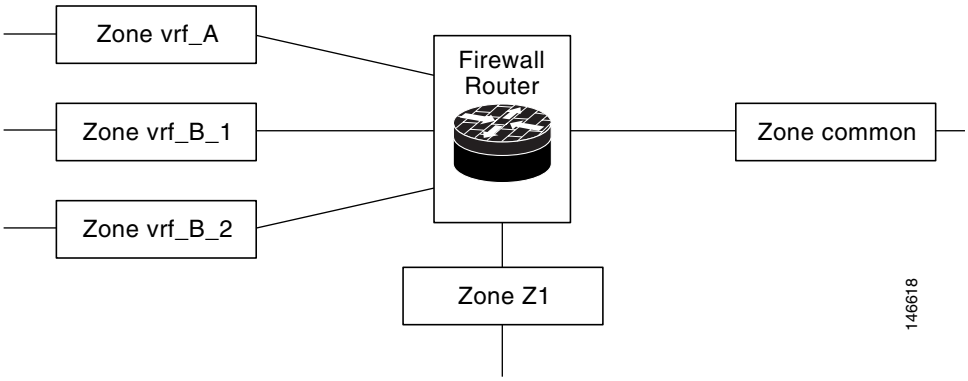
The Cisco IOS firewall is VPN routing and forwarding (VRF)-aware. It handles IP address overlap across different VRFs, separate thresholds and timeouts for VRFs, and so forth. All interfaces in a zone must belong to the same VRF.

However, you should not group interfaces from different VRFs in the same zone because VRFs belong to different entities that typically have their own policies.

You can configure a zone pair between two zones that contain different VRFs, as shown in Figure 3.

When multiple VRFs are configured on a router and an interface provides common services to all the VRFs (for example, internet service), you should place that interface in a separate zone. You can then define policies between the common zone and other zones. (There can be one or more zones per VRF.)

Figure 3 Zones and VRF



In Figure 3, the interface providing common services is a member of the zone “common.” All of VRF A is in a single zone, vrf_A. VRF B, which has multiple interfaces, is partitioned into multiple zones vrf_B_1 and vrf_B_2. Zone Z1 does not have VRF interfaces. You can specify policies between each of these zones and the common zone. Additionally, you can specify polices between each of the zones vrf_A, vrf_B_n, and Z1 if VRF route export is configured and the traffic patterns make sense. You can configure a policy between zones vrf_A and vrf_B_1, but be sure that traffic can flow between them.

There is no need to specify the global thresholds and timers on a per-VRF basis. Instead, parameters are supplied to the **inspect** action through a parameter map.

Zones and Transparent Firewalls

The Cisco IOS firewall supports transparent firewalls where the interfaces are placed in bridging mode and IP firewalling is performed on the bridged traffic.

To configure a transparent firewall, use the **bridge** command to enable the bridging of a specified protocol in a specified bridge and the **zone-member security** command to attach an interface to a zone. The **bridge** command on the interface indicates that the interface is in bridging mode.

A bridged interface can be a member of a zone. In a typical case, the Layer 2 domain is partitioned into zones and a policy is applied the same way as for Layer 3 interfaces.

Transparent Firewall Restriction for P2P Inspection

A Cisco IOS firewall uses network-based application recognition (NBAR) for peer-to-peer (P2P) protocol classification and policy enforcement. NBAR is not available for bridged packets; thus, all P2P packet inspection is not supported for firewalls with transparent bridging.

Overview of Security Zone Firewall Policies

A class is a way of identifying a set of packets based on its contents. Normally you define a class so that you can apply an action on the identified traffic that reflects a policy. A class is designated through class maps.

An action is a specific functionality. It typically is associated with a traffic class. For example, **inspect**, **drop**, and **pass** are actions.

To create firewall policies, you should complete the following tasks:

- Define a match criterion (class map)
- Associate actions to the match criteria (policy map)
- Attach the policy map to a zone pair (service policy)

The **class-map** command creates a class map to be used for matching packets to a specified class. Packets arriving at the targets (such as the input interface, output interface, or zone pair), determined by how the **service-policy** command is configured, are checked against the match criteria configured for a class map to determine if the packet belongs to that class.

The **policy-map** command creates or modifies a policy map that can be attached to one or more targets to specify a service policy. Use the **policy-map** command to specify the name of the policy map to be created, added to, or modified before you can configure policies for classes whose match criteria are defined in a class map.

Class Maps and Policy Maps for Zone-Based Policy Firewalls

Quality of service (QoS) class maps have numerous match criteria; firewalls have fewer match criteria. Firewall class maps have type **inspect**; this information controls what shows up under firewall class maps.

A policy is an association of traffic classes and actions. It specifies what actions should be performed on the defined traffic classes. An action is a specific function, and it is typically associated with a traffic class. For example, **inspect** and **drop** are actions.

Layer 3 and Layer 4 Class Maps and Policy Maps

Layer 3 and Layer 4 class maps are used to identify traffic streams on which different actions should be performed.

A Layer 3 or Layer 4 policy map is sufficient for the basic inspection of traffic.

The following example shows how to configure class map c1 with the match criteria of ACL 101 and the HTTP protocol, and create an inspect policy map named p1 to specify that packets will be dropped on the traffic at c1:

```
Router(config)# class-map type inspect match-all c1
Router(config-cmap)# match access-group 101
Router(config-cmap)# match protocol http

Router(config)# policy-map type inspect p1
Router(config-pmap)# class type inspect c1
Router(config-pmap-c)# drop
```

To create a Layer 3 or Layer 4 policy, see the "Configuring Layer 7 Protocol-Specific Firewall Policies"section.

Class-Map Configuration Restriction

If traffic meets multiple match criteria, the match criteria must be applied in the order of specific to less specific. For example, consider the following class map example:

```
class-map type inspect match-any my-test-cmap
 match protocol http
 match protocol tcp
```

In this example, HTTP traffic must first encounter the **match protocol http** command to ensure that the traffic will be handled by the service-specific capabilities of HTTP inspection. If the "match" lines were reversed so traffic encountered the **match protocol tcp** command before it was compared to the **match protocol http** command, the traffic would simply be classified as TCP traffic and inspected according to the capabilities of the Firewall's TCP Inspection component. This configuration would be a problem for services such as FTP and TFTP, and for several multimedia and voice signaling services such as H.323, Session Initiation Protocol (SIP), Skinny, and RTSP. These services require additional inspection capabilities to recognize their more complex activities.

Rate Limiting (Policing) Traffic Within a Layer 3 and Layer 4 Policy Map

In Cisco IOS Release 12.4(9)T, you can use the **police** command within an inspect policy to limit the number of concurrent connections allowed for applications such as Instant Messenger and P2P.

To effectively use the **police** command, you must also enable Cisco IOS stateful packet inspection within the inspect policy map. If you configure the **police** command without configuring the inspect action (through the **inspect** command), you will receive an error message and the **police** command will be rejected.

Compatibility with Existing Police Actions

Police actions provisioned in a modular quality of service (QoS) CLI (MQC) policy map are applied as input and output policies on an interface. An inspect policy map can be applied only to a zone pair, not to an interface. The police action will be enforced on traffic that traverses the zone pair. (The direction is inherent to the specification of the zone pair.) Thus, a QoS policy containing a police action can be present on interfaces that make up a zone pair and a police action can also be present in an inspect policy map applied across the zone pair. If both police actions are configured, the zone pair policer is executed after the input, interface policer, but before the output, interface policer. There is no interaction between the QoS and the inspect policers.

Police Restrictions

- The police action is not allowed in policies that are attached to zone pairs involving a “self” zone. Use Control Plane Policing if you want to perform this task.
- Policing can be specified only in Layer 3 and Layer 4 policy maps; it cannot be specified in Layer 7 policy maps.

Layer 7 Class Maps and Policy Maps

Layer 7 class maps can be used in inspect policy maps only for deep packet inspection (DPI).

To create a Layer 7 class map, use the **class-map type inspect** command for the desired protocol. For example, for the HTTP protocol you would enter the **class-map type inspect http** command.

The type of class map (for example, HTTP) determines the match criteria that you can use. For example, if you want to specify HTTP traffic that contains Java applets, you must specify a “match response body java” statement in the context of an “inspect HTTP” class map.

A Layer 7 policy map provides application-level inspection of traffic. The policy map can include class maps only of the same type.

The DPI functionality is delivered through Layer 7 class maps and policy maps.

To create a Layer 7 policy map, specify the protocol in the applicable **policy-map type inspect** command. For example, to create a Layer 7 HTTP policy map, use the **policy-map type inspect http** command. In that command there is an argument where you enter the HTTP policy-map name.

If you do not specify a protocol name (for example, you use the **policy-map type inspect** command), you will be creating a Layer 3 or Layer 4 policy map, which can only be an inspect type policy map.

A Layer 7 policy map must be contained in a Layer 3 or Layer 4 policy map; it cannot be attached directly to a target. To attach a Layer 7 policy map to a top-level policy map, use the **service-policy** (policy-map) command and specify the application name (that is, HTTP, IMAP, POP3, SMTP, or SUNRPC). The parent class for a Layer 7 policy should have an explicit match criterion that matched only one Layer 7 protocol before the policy is attached.

If the Layer 7 policy map is in a lower level, you must specify the **inspect** action at the parent level for a Layer 7 policy map.

Layer 7 Supported Protocols

You can create Layer 7 class maps and policy maps for the following protocols:

- America Online (AOL) Instant Messenger (IM) protocol
- eDonkey P2P protocol
- FastTrack traffic P2P protocol
- Gnutella Version 2 traffic P2P protocol
- H.323 VoIP Protocol Version 4
- HTTP—The protocol used by web browsers and web servers to transfer files, such as text and graphic files
- Internet Message Access Protocol (IMAP)—Method of accessing e-mail or bulletin board messages kept on a mail server that can be shared
- I Seek You (ICQ) IM protocol
- Kazaa Version 2 P2P protocol

- MSN Messenger IM protocol
- Post Office Protocol, Version 3 (POP3)—Protocol that client e-mail applications use to retrieve mail from a mail server
- SIP—Session Initiation Protocol (SIP)
- SMTP—Simple Network Management Protocol
- SUNRPC—Sun RPC (Remote Procedure Call)
- Windows Messenger IM Protocol
- Yahoo IM protocol

For information on configuring a Layer 7 class map and policy map (policies), see the "Configuring Layer 7 Protocol-Specific Firewall Policies" section.

Class-Default Class Map

In addition to user-defined classes, a system-defined class map named class-default represents all packets that do not match any of the user-defined classes in a policy. It always is the last class in a policy map.

You can define explicit actions for this group of packets. If you do not configure any actions for class-default in an inspect policy, the default action is **drop**.

Note For a class-default in an inspect policy, you can configure only **drop** action or **pass** action.

The following example shows how to use class-default in a policy map. In this example, HTTP traffic is dropped and the remaining traffic is inspected. Class map c1 is defined for HTTP traffic, and class-default is used for a policy map p1.

```
Router(config)# class-map type inspect match-all c1
Router(config-cmap)# match protocol http

Router(config)# policy-map type inspect p1
Router(config-pmap)# class type inspect c1
Router(config-pmap-c)# drop
Router(config-pmap)# class class-default
Router(config-pmap-c)# drop
```

Hierarchical Policy Maps

A policy can be nested within a policy. A policy that contains a nested policy is called a hierarchical policy.

To create a hierarchical policy, attach a policy directly to a class of traffic. A hierarchical policy contains a child and a parent policy. The child policy is the previously defined policy that is associated with the new policy through the use of the **service-policy** command. The new policy using the pre existing policy is the parent policy.

Note There can be a maximum of two levels in a hierarchical inspect service policy.

Parameter Maps

A parameter map allows you to specify parameters that control the behavior of actions and match criteria specified under a policy map and a class map, respectively.

There are three types of parameter maps:

- Inspect parameter map

  An inspect parameter map is optional. If you do not configure a parameter map, the software uses default parameters. Parameters associated with the inspect action apply to all nested actions (if any). If parameters are specified in both the top and lower levels, those in the lower levels override those in the top levels.

- URL Filter parameter map

  A parameter map is required for URL filtering (through the URL Filter action in a Layer 3 or Layer 4 policy map and the URL Filter parameter map).

- Protocol-specific parameter map

  A parameter map is required for an Instant Messenger application (Layer 7) policy map.

WAAS Support for the Cisco IOS Firewall

The WAAS firewall software, which was introduced in Cisco IOS Release 12.4(15)T, provides an integrated firewall that optimizes security-compliant WANs and application acceleration solutions with the following benefits:

- Optimizes a WAN through full stateful inspection capabilities
- Simplifies Payment Card Industry (PCI) compliance
- Protects transparent WAN accelerated traffic
- Integrates WAAS networks transparently
- Supports the Network Management Equipment (NME) WAE modules or standalone WAAS device deployment

WAAS has an automatic discovery mechanism that uses TCP options during the initial three-way handshake used to identify WAE devices transparently. After automatic discovery, optimized traffic flows (paths) experience a change in the TCP sequence number to allow endpoints to distinguish between optimized and nonoptimized traffic flows.

Note Paths are synonymous with connections.

WAAS allows the Cisco IOS firewall to automatically discover optimized traffic by enabling the sequence number to change without compromising the stateful Layer 4 inspection of TCP traffic flows that contain internal firewall TCP state variables. These variables are adjusted for the presence of WAE devices.

If the Cisco IOS firewall notices that a traffic flow has successfully completed WAAS automatic discovery, it permits the initial sequence number shift for the traffic flow and maintains the Layer 4 state on the optimized traffic flow.

Note Stateful Layer 7 inspection on the client side can also be performed on nonoptimized traffic.

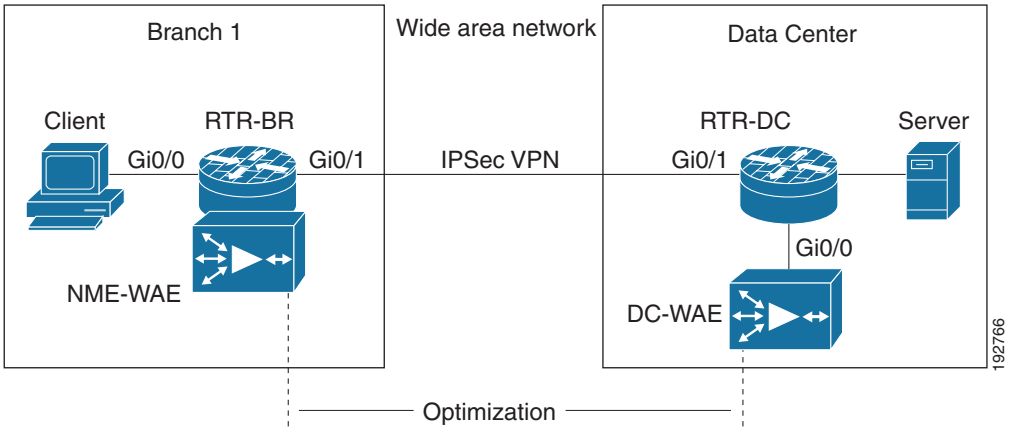
WAAS Traffic Flow Optimization Deployment Scenarios

The following sections describe three different WAAS traffic flow optimization scenarios for branch office deployments. WAAS traffic flow optimization works with the Cisco IOS firewall feature on a Cisco Integrated Services Router (ISR).

- WAAS Branch Deployment with an Off-Path Device, page 13
- WAAS Branch Deployment with an Inline Device, page 14

Figure 4 shows an example of an end-to-end WAAS traffic flow optimization with the Cisco IOS firewall. In this particular deployment, an NME-WAE device is on the same router as the Cisco IOS firewall. WCCP is used to redirect traffic for interception.

Figure 4 End-to-End WAAS Optimization Path

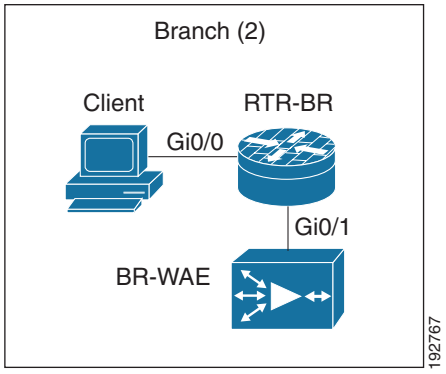


WAAS Branch Deployment with an Off-Path Device

A WAE device can be either an NME-WAE that is installed on an ISR as an integrated service engine (as shown in Figure 4) or a standalone WAE device.

Figure 5 shows a WAAS branch deployment that uses WCCP to redirect traffic to an off-path, standalone WAE device for traffic interception. The configuration for this option is the same as the WAAS branch deployment with an NME-WAE.

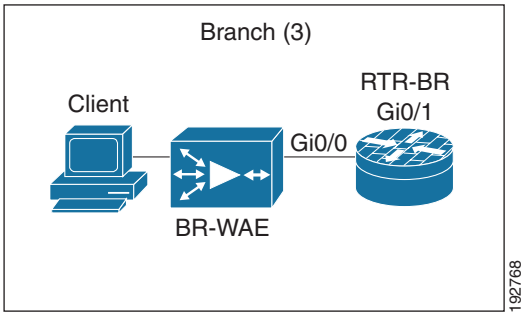
Figure 5 WAAS Off-Path Branch Deployment



WAAS Branch Deployment with an Inline Device

Figure 6 shows a WAAS branch deployment that has an inline WAE device that is physically in front of the ISR router. Because the WAE device is in front of the router, Layer 7 inspection on the client side is not supported because the Cisco IOS firewall receives WAAS optimized packets.

Figure 6 WAAS Inline Path Branch Deployment



An edge WAAS device with the Cisco IOS firewall is applied at branch office sites that must inspect traffic moving to and from a WAN connection. The Cisco IOS firewall monitors traffic for optimization indicators (TCP options and subsequent TCP sequence number changes) and allows optimized traffic to pass, while still applying Layer 4 stateful inspection and deep packet inspection to all traffic, maintaining security while accommodating WAAS optimization advantages.

Note If the WAE device is in the inline location, the device enters its bypass mode after the automatic discovery process. Although the router is not directly involved in WAAS optimization, the router must be aware that WAAS optimization is applied to the traffic in order to apply the Cisco IOS firewall inspection to network traffic and make allowances for optimization activity if optimization indicators are present.

Out-of-Order Packet Processing Support in the Zone-Based Firewall Application

Out-of-Order (OoO) packet processing support for Common Classification Engine (CCE) firewall application and CCE adoptions of the Intrusion Prevention System (IPS) allows for packets that arrive out of order to be copied and reassembled in the correct order. This enhancement reduces the need to retransmit dropped packets and reduces the bandwidth needed for transmission on a network. To configure OoO support, use the **parameter-map type ooo global** command.

Note IPS sessions use OoO parameters configured using the **parameter-map type ooo global** command.

Note OoO processing is not supported in SMTP because SMTP supports masking actions that require packet modification.

OoO packet processing support is enabled by default when a Layer 7 policy is configured for Deep Packet Inspection (DPI) for the following protocols:

- AOL IM protocol
- eDonkey P2P protocol
- FastTrack traffic P2P protocol
- Gnutella Version 2 traffic P2P protocol
- H.323 VoIP Protocol Version 4
- HTTP—The protocol used by web browsers and web servers to transfer files, such as text and graphic files
- IMAP—Method of accessing e-mail or bulletin board messages kept on a mail server that can be shared
- ICQ IM Protocol
- Kazaa Version 2 P2P protocol
- Match Protocol SIP—Match Protocol Session Initiation Protocol (SIP)
- MSN Messenger IM protocol
- POP3—Protocol that client e-mail applications use to retrieve mail from a mail server
- SUNRPC—Sun RPC (Remote Procedure Call)
- Windows Messenger IM Protocol
- Yahoo IM protocol

For information on configuring a Layer 7 class map and policy map (policies), see the "Configuring Layer 7 Protocol-Specific Firewall Policies" section.

Note OoO packets are dropped when Cisco IOS Intrusion Prevention System (IPS) and zone-based firewall with L4 inspection are enabled.

Intrazone Support in the Zone-Based Firewall Application

Intrazone support allows for zone configuration to include users both inside and outside a network. This allows for traffic inspection between users belonging to the same zone but different networks. Before Cisco IOS Release 15.0(1)M, traffic within a zone was allowed to pass uninspected by default. To configure a zone pair definition with the same zone for source and destination, use the **zone-pair security** command. This allows the functionality of attaching a policy map and inspecting the traffic within the same zone.

How to Configure Zone-Based Policy Firewall

- Configuring Layer 3 and Layer 4 Firewall Policies, page 16 (required)
- Configuring a Parameter Map, page 19 (required)
- Configuring OoO Packet Processing Support in the Zone-Based Firewall Applications, page 25 (optional)
- Configuring Intrazone Support in the Zone-Based Firewall Applications, page 27 (optional)
- Configuring Layer 7 Protocol-Specific Firewall Policies, page 28 (optional)

- Creating Security Zones and Zone Pairs, and Attaching a Policy Map to a Zone Pair, page 51 (required)
- Configuring the Cisco IOS Firewall with WAAS, page 54(optional)

Configuring Layer 3 and Layer 4 Firewall Policies

Layer 3 and Layer 4 policies are "top level" policies that are attached to the target (zone pair). Use the following tasks to configure Layer 3 and Layer 4 firewall policies:

- Configuring a Class Map for a Layer 3 and Layer 4 Firewall Policy, page 16
- Creating a Policy Map for a Layer 3 and Layer 4 Firewall Policy, page 17

Configuring a Class Map for a Layer 3 and Layer 4 Firewall Policy

Use this task to configure a class map for classifying network traffic.

Note You must perform at least one match step from Step 4, 5, or 6.

When packets are matched to an access group, protocol, or class map, a traffic rate is generated for these packets. In a zone-based firewall policy, only the first packet that creates a session matches the policy. Subsequent packets in this flow do not match the filters in the configured policy, but instead match the session directly. The statistics related to subsequent packets are shown as part of the inspect action.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** [**match-any** | **match-all**] *class-map-name*
4. **match access-group** {*access-group* | **name** *access-group-name*}
5. **match protocol** *protocol-name* [**signature**]
6. **match class-map** *class-map-name*
7. **end**
8. **show policy-map type inspect zone-pair session**

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	**class-map type inspect** [**match-any** | **match-all**] *class-map-name* **Example:** `Router(config)# class-map type inspect match-all c1`	Creates a Layer 3 or Layer 4 inspect type class map and enters QoS class-map configuration mode.
Step 4	**match access-group** {*access-group* | **name** *access-group-name*} **Example:** `Router(config-cmap)# match access-group 101`	Configures the match criteria for a class map based on the ACL name or number.
Step 5	**match protocol** *protocol-name* [**signature**] **Example:** `Router(config-cmap)# match protocol http`	Configures the match criteria for a class map on the basis of a specified protocol. • Only Cisco IOS stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps. • **signature**—Signature-based classification for peer-to-peer (P2P) packets is enabled.
Step 6	**match class-map** *class-map-name* **Example:** `Router(config-cmap)# match class-map c1`	Specifies a previously defined class as the match criteria for a class map.
Step 7	**exit** **Example:** `Router(config-cmap)# end`	Returns to privileged EXEC mode.
Step 8	**show policy-map type inspect zone-pair session** **Example:** `Router(config-cmap)# show policy-map type inspect zone-pair session`	(Optional) Displays the Cisco IOS stateful packet inspection sessions created because of the policy-map application on the specified zone pair. **Note** The information shown under the class-map field is the traffic rate (bits per second) of the traffic belonging to the connection initiating traffic only. Unless the connection setup rate is significantly high and sustained for multiple intervals over which the rate is computed, no significant data is shown for the connection.

Creating a Policy Map for a Layer 3 and Layer 4 Firewall Policy

Use this task to create a policy map for a Layer 3 and Layer 4 firewall policy that will be attached to zone pairs.

Note If you are creating an inspect type policy map, note that only the following actions are allowed: drop, inspect, police, pass, service-policy, and urlfilter.

Note You must perform at least one step from Step 5, 8, 9, or 10.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class type inspect** *class-name*
5. **inspect** [*parameter-map-name*]
6. **police rate** *bps* **burst** *size*
7. **drop** [**log**]
8. **pass**
9. **service-policy type inspect** *policy-map-name*
10. **urlfilter** *parameter-map-name*
11. **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	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect p1`	Creates a Layer 3 and Layer 4 inspect type policy map and enters QoS policy-map configuration mode.
Step 4	**class type inspect** *class-name* **Example:** `Router(config-pmap)# class type inspect c1`	Specifies the traffic (class) on which an action is to be performed.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Router(config-pmap-c)# inspect inspect-params`	Enables Cisco IOS stateful packet inspection.

	Command or Action	Purpose
Step 6	**police rate** *bps* **burst** *size* **Example:** `Router(config-pmap-c)# police rate 2000 burst 3000`	(Optional) Limits traffic matching within a firewall (inspect) policy.
Step 7	**drop** [**log**] **Example:** `Router(config-pmap-c)# drop`	(Optional) Drops packets that are matched with the defined class. **Note** The actions **drop** and **pass** are exclusive, and the actions **inspect** and **drop** are exclusive; that is, you cannot specify both of them.
Step 8	**pass** **Example:** `Router(config-pmap-c)# pass`	(Optional) Allows packets that are matched with the defined class.
Step 9	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-pmap-c)# service-policy type inspect p1`	Attaches a firewall policy map to a zone pair.
Step 10	**urlfilter** *parameter-map-name* **Example:** `Router(config-pmap-c)# urlfilter param1`	(Optional) Enables Cisco IOS firewall URL filtering.
Step 11	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to QoS policy-map configuration mode.

Configuring a Parameter Map

Depending on your policy, you can configure either an inspect, URL filter, or protocol-specific type parameter map. If you are configuring a URL filter type or protocol-specific type policy, you must configure a parameter map, as appropriate. However, a parameter map is optional if you are using an inspect type policy.

Note Changes to the parameter map are not reflected on connections already established through the firewall. Changes are applicable only to new connections permitted to the firewall. To ensure that your firewall enforces policies strictly, clear all the connections allowed in the firewall after you change the parameter map. To clear existing connections, use the **clear zone-pair inspect sessions** command.

Use one of the following tasks to configure a parameter map:

- Creating an Inspect Parameter Map, page 20
- Creating a URL Filter Parameter Map, page 22
- Configuring a Layer 7 Protocol-Specific Parameter Map, page 24

Creating an Inspect Parameter Map

Use this task to create an inspect type parameter map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** {*parameter-map-name* | **global** | **default**}
4. **log** {**dropped-packets** {**disable** | **enable**} | **summary** [**flows** *number*] [**time-interval** *seconds*]}
5. **alert** {**on** | **off**}
6. **audit-trail** {**on** | **off**}
7. **dns-timeout** *seconds*
8. **icmp idle-timeout** *seconds*
9. **max-incomplete** {**low** | **high**} *number-of-connections*
10. **one-minute** {**low** | **high**} *number-of-connections*
11. **sessions maximum** *sessions*
12. **tcp finwait-time** *seconds*
13. **tcp idle-time** *seconds*
14. **tcp max-incomplete host** *threshold* [**block-time** *minutes*]
15. **tcp synwait-time** *seconds*
16. **tcp window-scale-enforcement loose**
17. **udp idle-time** *seconds*
18. **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	**parameter-map type inspect** {*parameter-map-name* | **global** | **default**} **Example:** `Router(config)# parameter-map type inspect eng-network-profile`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action, and enters parameter map type inspect configuration mode.

	Command or Action	Purpose
Step 4	**log** {**dropped-packets** {**disable** | **enable**} | **summary** [**flows** *number*] [**time-interval** *seconds*]} **Example:** `Router(config-profile)# log summary flows 15 time-interval 30`	(Optional) Configures packet logging during the firewall activity. **Note** This command is visible in the global parameter map type inspect configuration mode only.
Step 5	**alert** {**on** | **off**} **Example:** `Router(config-profile)# alert on`	(Optional) Turns on and off Cisco IOS stateful packet inspection alert messages that are displayed on the console.
Step 6	**audit-trail** {**on** | **off**} **Example:** `Router(config-profile)# audit-trail on`	(Optional) Turns audit trail messages on or off.
Step 7	**dns-timeout** *seconds* **Example:** `Router(config-profile)# dns-timeout 60`	(Optional) Specifies the domain name system (DNS) idle timeout (the length of time for which a DNS lookup session will continue to be managed while there is no activity).
Step 8	**icmp idle-timeout** *seconds* **Example:** `Router(config-profile)# icmp idle-timeout 90`	(Optional) Configures the timeout for Internet Control Message Protocol (ICMP) sessions.
Step 9	**max-incomplete** {**low** | **high**} {*number-of-connections*} **Example:** `Router(config-profile)# max-incomplete low 800`	(Optional) Defines the number of existing half-open sessions that will cause the Cisco IOS firewall to start and stop deleting half-open sessions.
Step 10	**one-minute** {**low** | **high**} *number-of-connections* **Example:** `Router(config-profile)# one-minute low 300`	(Optional) Defines the number of new unestablished sessions that will cause the system to start deleting half-open sessions and stop deleting half-open sessions.
Step 11	**sessions maximum** *sessions* **Example:** `Router(config-profile)# sessions maximum 200`	(Optional) Sets the maximum number of allowed sessions that can exist on a zone pair. • You may want to use this command to limit the bandwidth used by the sessions. • *sessions*—Maximum number of allowed sessions. Range: 1 to 2147483647.
Step 12	**tcp finwait-time** *seconds* **Example:** `Router(config-profile)# tcp finwait-time 5`	(Optional) Specifies how long a TCP session will be managed after the Cisco IOS firewall detects a FIN-exchange.
Step 13	**tcp idle-time** *seconds* **Example:** `Router(config-profile)# tcp idle-time 90`	(Optional) Configures the timeout for TCP sessions.

	Command or Action	Purpose
Step 14	**tcp max-incomplete host** *threshold* [**block-time** *minutes*] **Example:** `Router(config-profile)# tcp max-incomplete host 500 block-time 10`	(Optional) Specifies threshold and blocking time values for TCP host-specific DoS detection and prevention.
Step 15	**tcp synwait-time** *seconds* **Example:** `Router(config-profile)# tcp synwait-time 3`	(Optional) Specifies how long the software will wait for a TCP session to reach the established state before dropping the session.
Step 16	**tcp window-scale-enforcement loose** **Example:** `Router(config-profile)# tcp window-scale-en-forcement loose`	(Optional) Disables the window scale option check in the parameter map for a TCP packet that has an invalid window scale option under the Zone Based Firewall (ZBF).
Step 17	**udp idle-time** *seconds* **Example:** `Router(config-profile)# udp idle-time 75`	(Optional) Configures the idle timeout of UDP sessions going through the firewall.
Step 18	**exit** **Example:** `Router(config-profile)# exit`	Returns to global configuration mode.

Creating a URL Filter Parameter Map

To create a URL filter parameter map, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type urlfilter** *parameter-map-name*
4. **alert** {**on** | **off**}
5. **allow-mode** {**on** | **off**}
6. **audit-trail** {**on** | **off**}
7. **cache** *number*
8. **exclusive-domain** {**deny** | **permit**} *domain-name*
9. **max-request** *number-of-requests*
10. **max-resp-pak** *number-of-requests*
11. **server vendor** {**n2h2** | **websense**} {*ip-address* | *hostname* [**port** *port-number*]} [**outside**] [**log**] [**retrans** *retransmission-count*] [**timeout** *seconds*]
12. **source-interface** *interface-name*

13. **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	**parameter-map type urlfilter** *parameter-map-name* **Example:** `Router(config)# parameter-map type urlfilter eng-network-profile`	Creates or modifies a parameter map for URL filtering parameters and enters parameter map inspect configuration mode. **Note** This command is hidden in releases later than Cisco IOS Release 12.4(20)T, but it continues to work. The **parameter-map type urlfpolicy** command can also be used. This command is used to create URL filtering parameters for local, trend, Websense Internet filtering, and the N2H2 Internet blocking program. We recommend the use of the URL filter policy rather than the URL filter action for Cisco IOS Release 12.4(20)T. All the use cases supported by the URL filter as an action are also supported by the URL filter policy. See the "Configuring a URL Filter Policy" section on page 34 for more information.
Step 4	**alert** {**on** | **off**} **Example:** `Router(config-profile)# alert on`	(Optional) Turns on or off Cisco IOS stateful packet inspection alert messages that are displayed on the console.
Step 5	**allow-mode** {**on** | **off**} **Example:** `Router(config-profile)# allow-mode on`	(Optional) Turns on or off the default mode of the filtering algorithm.
Step 6	**audit-trail** {**on** | **off**} **Example:** `Router(config-profile)# audit-trail on`	(Optional) Turns audit trail messages on or off.
Step 7	**cache** *number* **Example:** `Router(config-profile)# cache 5`	(Optional) Controls how the URL filter handles the cache it maintains of HTTP servers.

	Command or Action	Purpose
Step 8	**exclusive-domain** {**deny** | **permit**} *domain-name* **Example:** `Router(config-profile)# exclusive-domain permit cisco.com`	(Optional) Adds or removes a domain name to or from the exclusive domain list so that the Cisco IOS firewall does not have to send lookup requests to the vendor server.
Step 9	**max-request** *number-of-requests* **Example:** `Router(config-profile)# max-request 80`	(Optional) Specifies the maximum number of outstanding requests that can exist at a time.
Step 10	**max-resp-pak** *number-of-requests* **Example:** `Router(config-profile)# max-resp-pak 200`	(Optional) Specifies the maximum number of HTTP responses that the Cisco IOS firewall can keep in its packet buffer.
Step 11	**server vendor** {**n2h2** | **websense**} {*ip-address* | *hostname* [**port** *port-number*]} [**outside**] [**log**] [**retrans** *retransmission-count*] [**timeout** *seconds*] **Example:** `Router(config-profile)# server vendor n2h2 10.193.64.22 port 3128 outside retrans 9 timeout 8`	Specifies the URL filtering server. **Note** This command is mandatory if you want anything from the URL filter configuration.
Step 12	**source-interface** *interface-name* **Example:** `Router(config-profile)# source-interface ethernet0`	(Optional) Specifies the interface whose IP address is used as the source IP address while making a TCP connection to the URL filter server (Websense or N2H2).
Step 13	**exit** **Example:** `Router(config-profile)# exit`	Returns to global configuration mode.

Configuring a Layer 7 Protocol-Specific Parameter Map

Use this task to configure a Layer 7, protocol-specific parameter map.

Note Protocol-specific parameter maps can be created only for instant messenger applications (AOL, ICQ, MSN Messenger, Yahoo Messenger, and Windows Messenger).

Prerequisites

To enable name resolution to occur, you must also enable the **ip domain name** command and the **ip name-serve**r command.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **parameter-map type protocol-info** *parameter-map-name*
4. **server** {**name** *string* [**snoop**] | **ip** {*ip-address* | **range** *ip-address-start ip-address-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	**parameter-map type protocol-info** *parameter-map-name* **Example:** `Router(config)# parameter-map type protocol-info ymsgr`	Defines an application-specific parameter map and parameter map type inspect configuration mode. **Note** Protocol-specific parameter maps can be created only for instant messenger applications (AOL, ICQ, MSN Messenger, Yahoo Messenger, and Windows Messenger).
Step 4	**server** {**name** *string* [**snoop**] | **ip** {*ip-address* | **range** *ip-address-start ip-address-end*}} **Example:** `Router(config-profile)# server name sdsc.msg.example.com`	Configures a set of Domain Name System (DNS) servers for which a given instant messenger application will be interacting. **Note** If at least one server instance is not configured, the parameter map will not have any definitions to enforce; that is, the configured instant messenger policy cannot be enforced. **Note** To configure more than one set of servers, you can issue the **server** command multiple times within an instant messenger's parameter map. Multiple entries are treated cumulatively.

Troubleshooting Tips

To display details of an IM protocol-specific parameter map, use the **show parameter-map type protocol-info** command.

Configuring OoO Packet Processing Support in the Zone-Based Firewall Applications

Use this task to configure OoO packet processing support in zone-based firewall applications.

Note If a TCP-based Layer 7 policy is configured for DPI, OoO is enabled by default. Use the **parameter-map type ooo global** command to configure the OoO packet support parameters or to turn off OoO processing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type ooo global**
4. **tcp reassembly alarm** {**on** | **off**}
5. **tcp reassembly memory limit** *memory-limit*
6. **tcp reassembly queue length** *queue-length*
7. **tcp reassembly timeout** *time-limit*
8. **exit**

Note In Cisco IOS 12.4(15)T, OoO processing was enabled for zone-based firewall and Intrusion Prevention Systems (IPS) shared sessions with L4 match as match protocol, TCP match protocol http, or any TCP based L7 expecting packet ordering.

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	**parameter-map type ooo global** **Example:** `Router(config)# parameter-map type ooo global`	Enters parameter map configuration mode.
Step 4	**tcp reassembly alarm** {**on** | **off**} **Example:** `Router(config-profile)# tcp reassembly alarm on`	Specifies the alert message configuration.
Step 5	**tcp reassembly memory limit** *memory-limit* **Example:** `Router(config-profile)# tcp reassembly memory limit 2048`	Specifies the OoO box-wide buffer size.
Step 6	**tcp reassembly queue length** *queue-length* **Example:** `Router(config-profile)# tcp reassembly queue length 45`	Specifies OoO queue length per TCP flow.

	Command or Action	Purpose
Step 7	**tcp reassembly timeout** *time-limit* **Example:** `Router(config-profile)# tcp reassembly timeout 34`	Specifies the OoO queue reassembly timeout value.
Step 8	**exit** **Example:** `Router(config-profile)# exit`	Returns to global configuration mode.

Configuring Intrazone Support in the Zone-Based Firewall Applications

Use this task to configure intrazone support when using a zone-based firewall.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone-pair security** *zone-pair-name* [**source** *source-zone-name* **destination** *destination-zone-name*]
4. **policy-map type inspect** *policy-map-name*
5. **class type inspect** *protocol-type class-map-name*
6. **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	**zone-pair security** *zone-pair-name* [**source** *source-zone-name* **destination** *destination-zone-name*] **Example:** `Router(config)# zone-pair security zonepair17 source zone8 destination zone8`	Specifies the name of the zone pair being attached to an interface, the source zone for information, and the destination zone for information passing through this zone pair. **Note** To configure intrazone support, the source zone and the destination zone must be the same.
Step 4	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect my-pmap`	Specifies the policy map name and enters Qos policy-map configuration mode.

	Command or Action	Purpose
Step 5	**class type inspect** *protocol-name class-map-name* **Example:** `Router(config-pmap)# class type inspect aol cmap1`	Specifies the firewall class map protocol and name.
Step 6	**exit** **Example:** `Router(config)# exit`	Returns to global configuration mode.

Configuring Layer 7 Protocol-Specific Firewall Policies

Configure Layer 7 policy maps if you are interested in extra provisioning for Layer 7 inspection modules. It is not necessary that you configure all of the Layer 7 policy maps.

Use one of the following tasks to configure a Layer 7, protocol-specific firewall policy:

- Configuring an HTTP Firewall Policy, page 29 (optional)
- Configuring a URL Filter Policy, page 34 (optional)
- Configuring an IMAP Firewall Policy, page 35 (optional)
- Configuring an Instant Messenger Policy, page 37 (optional)
- Configuring a Peer-to-Peer Policy, page 39 (optional)
- Configuring a POP3 Firewall Policy, page 42 (optional)
- Configuring an SMTP Firewall Policy, page 44 (optional)
- Configuring a SUNRPC Firewall Policy, page 46 (optional)
- Configuring an MSRPC Firewall Policy, page 48 (optional)

Layer 7 Class Map and Policy Map Restrictions

- DPI class maps for Layer 7 can be used in inspect policy maps of the respective type. For example, **class-map type inspect http** can only be used only in **policy-map type inspect http**.
- DPI policies require an **inspect** action at the parent level.
- A Layer 7 (DPI) policy map must be nested at the second level in a Layer 3 or Layer 4 inspect policy map, whereas a Layer 3 or Layer 4 inspect policy can be attached at the first level. Therefore, a Layer 7 policy map cannot be attached directly to a zone pair.
- If no action is specified in the hierarchical path of an inspect service policy, the packet is dropped. Traffic matching class-default in the top-level policy is dropped if there are no explicit actions configured in class-default. If the traffic does not match any class in a Layer 7 policy, the traffic is not dropped; control returns to the parent policy and subsequent actions (if any) in the parent policy are executed on the packet.
- Layer 7 policy maps include class maps only of the same type.
- You can specify the **reset** action only for TCP traffic; it resets the TCP connection.

- In Cisco IOS Release 15.1(4)M and later releases, removing a class that has a header with a regular expression from a Layer 7 policy map causes active HTTP sessions to reset. Prior to this change, when a class was removed from a Layer 7 policy map, the router reloaded.

Configuring an HTTP Firewall Policy

Use the following tasks to configure an HTTP firewall policy:

- Configuring an HTTP Firewall Class Map, page 29
- Configuring an HTTP Firewall Policy Map, page 33

If you want to configure match criteria on the basis of an element within a parameter map, you must configure a parameter map as shown in the task "Creating an Inspect Parameter Map."

You must specify at least one match criterion; otherwise, the firewall policy will not be effective.

Configuring an HTTP Firewall Class Map

Use this task to configure an HTTP firewall class map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect http** [**match-any** | **match-all**] *class-map-name*
4. **match response body java-applet**
5. **match req-resp protocol violation**
6. **match req-resp body length** {**lt** | **gt**} *bytes*
7. **match req-resp header content-type** {**violation** | **mismatch** | **unknown**}
8. **match** {**request** | **response** | **req-resp**} **header** [*header-name*] **count gt** *number*
9. **match** {**request** | **response** | **req-resp**} **header** [*header-name*] **length gt** *bytes*
10. **match request** {**uri** | **arg**} **length gt** *bytes*
11. **match request method** {**connect** | **copy** | **delete** | **edit** | **get** | **getattribute** | **getattributenames** | **getproperties** | **head** | **index** | **lock** | **mkdir** | **move** | **options** | **post** | **put** | **revadd** | **revlabel** | **revlog** | **revnum** | **save** | **setattribute** | **startrev** | **stoprev** | **trace** | **unedit** | **unlock**}
12. **match request port-misuse** {**im** | **p2p** | **tunneling** | **any**}
13. **match req-resp header transfer-encoding** {**chunked** | **compress** | **deflate** | **gzip** | **identity** | **all**}
14. **match** {**request** | **response** | **req-resp**} **header** [*header-name*] **regex** *parameter-map-name*
15. **match request uri regex** *parameter-map-name*
16. **match** {**request** | **response** | **req-resp**} **body regex** *parameter-map-name*
17. **match response status-line regex** *parameter-map-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	**class-map type inspect http** [**match-any** | **match-all**] *class-map-name* **Example:** `Router(config)# class-map type inspect http http-class`	Creates a class map for the HTTP protocol so that you can enter match criteria and enters QoS class- map configuration mode.
Step 4	**match response body java-applet** **Example:** `Router(config-cmap)# match response body java-applet`	(Optional) Identifies Java applets in an HTTP connection.
Step 5	**match req-resp protocol violation** **Example:** `Router(config-cmap)# match req-resp protocol violation`	(Optional) Configures an HTTP class map to allow HTTP messages to pass through the firewall or to reset the TCP connection when HTTP noncompliant traffic is detected.
Step 6	**match req-resp body length** {**lt** | **gt**} *bytes* **Example:** `Router(config-cmap)# match req-resp body length gt 35000`	(Optional) Configures an HTTP class map to use the minimum or maximum message size, in bytes, as a match criterion for permitting or denying HTTP traffic through the firewall. • The number of bytes can be from 0 to 65535.
Step 7	**match req-resp header content-type** {**violation** | **mismatch** | **unknown**} **Example:** `Router(config-cmap)# match req-resp header content-type mismatch`	(Optional) This command configures an HTTP class map based on the content type of HTTP traffic.
Step 8	**match** {**request** | **response** | **req-resp**} **header** [*header-name*] **count gt** *number* **Example:** `Router(config-cmap)# match req-resp header count gt 16`	(Optional) Configure an HTTP firewall policy to permit or deny HTTP traffic on the basis of request, response, or both request and response messages whose header count does not exceed a maximum number of fields.

	Command or Action	Purpose
Step 9	**match** {**request** | **response** | **req-resp**} **header** [*header-name*] **length gt** *bytes* **Example:** `Router(config-cmap)# match response header length gt 50000`	(Optional) Permits or denies HTTP traffic based on the length of the HTTP request header. • *header-name*—Specific line in the header field. If a specific line is defined, only that specific field length will be used as a match criterion. • **gt** *bytes*—Maximum number of bytes that can be in the header of the HTTP request. Number of bytes range: 0 to 65535.
Step 10	**match request** {**uri** | **arg**} **length gt** *bytes* **Example:** `Router(config-cmap)# match request uri length gt 500`	(Optional) Configures an HTTP firewall policy to use the URI or argument length in the request message as a match criterion for permitting or denying HTTP traffic.
Step 11	**match request method** {**connect** | **copy** | **delete** | **edit** | **get** | **getattribute** | **getattributenames** | **getproperties** | **head** | **index** | **lock** | **mkdir** | **move** | **options** | **post** | **put** | **revadd** | **revlabel** | **revlog** | **revnum** | **save** | **setattribute** | **startrev** | **stoprev** | **trace** | **unedit** | **unlock**} **Example:** `Router(config-cmap)# match request method connect`	(Optional) Configures an HTTP firewall policy to use the request methods or the extension methods as a match criterion for permitting or denying HTTP traffic.
Step 12	**match request port-misuse** {**im** | **p2p** | **tunneling** | **any**} **Example:** `Router(config-cmap)# match request port-misuse any`	(Optional) Identifies applications misusing the HTTP port.
Step 13	**match req-resp header transfer-encoding** {**chunked** | **compress** | **deflate** | **gzip** | **identity** | **all**} **Example:** `Router(config-cmap)# match req-resp header transfer-encoding compress`	(Optional) Permits or denies HTTP traffic according to the specified transfer encoding of the message. • **chunked**—Encoding format (specified in RFC 2616, *Hypertext Transfer Protocol—HTTP/1*) in which the body of the message is transferred in a series of chunks; each chunk contains its own size indicator. • **compress**—Encoding format produced by the UNIX compress utility. • **deflate**—ZLIB format defined in RFC 1950, *ZLIB Compressed Data Format Specification Version 3.3*, combined with the deflate compression mechanism described in RFC 1951, *DEFLATE Compressed Data Format Specification Version 1.3*. • **gzip**—Encoding format produced by the gzip (GNU zip) program. • **identity**—Default encoding, which indicates that no encoding has been performed. • **all**—All of the transfer encoding types.

	Command or Action	Purpose
Step 14	**match** {**request** | **response** | **req-resp**} **header** [*header-name*] **regex** *parameter-map-name* **Example:** `Router(config-cmap)# match req-resp header regex non_ascii_regex`	(Optional) Configures HTTP firewall policy match criteria on the basis of headers that match the regular expression defined in a parameter map. • HTTP has two regular expression (regex) options. One combines the **header** keyword, **content-type** header name, and **regex** keyword and *parameter-map-name* argument. The other combines the **header** keyword and **regex** keyword and *parameter-map-name* argument. • If the **header** and **regex** keywords are used with the *parameter-map-name* argument, the parameter map does not require a period and asterisk in front of the *parameter-map-name* argument. For example, either "html" or ".*html" *parameter-map-name* argument can be configured. • If the **header** keyword is used with the content-type header name and **regex** keyword, then the parameter map name requires a period and asterisk (.*) in front of the *parameter-map-name* argument. For example, the *parameter-map-name* argument "html" is expressed as: .*html. **Note** If the period and asterisk are added in front of "html" (.*html), the *parameter-map-name* argument works for both HTTP regex options. • The **mismatch** keyword is valid only for the **match response header content-type regex** command syntax for messages that need to be matched that have a content-type header name mismatch. **Tip** It is a good practice to add ".*" to the **regex** *parameter-map-name* arguments that are not present at the beginning of a text string.
Step 15	**match request uri regex** *parameter-map-name* **Example:** `Router(config-cmap)# match request uri regex uri_regex_cm`	(Optional) Configures an HTTP firewall policy to permit or deny HTTP traffic on the basis of request messages whose URI or arguments (parameters) match a defined regular expression.
Step 16	**match** {**request** | **response** | **req-resp**} **body regex** *parameter-map-name* **Example:** `Router(config-cmap)# match response body regex body_regex`	(Optional) Configures a list of regular expressions that are to be matched against the body of the request, response, or both the request and response message.
Step 17	**match response status-line regex** *parameter-map-name* **Example:** `Router(config-cmap)# match response status-line regex status_line_regex`	(Optional) Specifies a list of regular expressions that are to be matched against the status line of a response message.

Configuring an HTTP Firewall Policy Map

Use this task to configure an HTTP firewall policy map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect http** *policy-map-name*
4. **class-type inspect http** *http-class-name*
5. **allow**
6. **log**
7. **reset**

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	**policy-map type inspect http** *policy-map-name* **Example:** `Router(config)# policy-map type inspect http myhttp-policy`	Creates a Layer 7 HTTP policy map and enters QoS policy-map configuration mode.
Step 4	**class-type inspect http** *http-class-name* **Example:** `Router(config-pmap)# class-type inspect http http-class`	Creates a class map for the HTTP protocol.
Step 5	**allow** **Example:** `Router(config-pmap)# allow`	(Optional) Allows a traffic class matching the class.

	Command or Action	Purpose
Step 6	**log** **Example:** `Router(config-pmap)# log`	Generates a log (messages).
Step 7	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Resets a TCP connection if the data length of the SMTP body exceeds the value configured in the **class-map type inspect smtp** command.

Configuring a URL Filter Policy

Use this task to configure a URL filter policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type urlfpolicy** {**local** | **n2h2** | **websense**} *parameter-map-name*
4. **exit**
5. **class-map type urlfilter** {*class-map-name* | **match-any** *class-map-name* | **n2h2** {*class-map-name* | **match-any** *class-map-name*} | **websense** {*class-map-name* | **match-any** *class-map-name*}}
6. **exit**
7. **policy-map type inspect urlfilter** *policy-map-name*
8. **service-policy urlfilter** *policy-map-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	**parameter-map type urlfpolicy** {**local** | **n2h2** | **websense**} *parameter-map-name* **Example:** `Router(config)# parameter-map type urlfpolicy websense websense-param-map`	Configures the URL filter name related to the parameter map, which can include the local, Websense, or N2H2 parameter and enters parameter map inspect configuration mode.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config-profile)# exit`	Exits parameter map type inspect configuration mode.
Step 5	**class-map type urlfilter** {*class-map-name* | **match-any** *class-map-name* | **n2h2** {*class-map-name* | **match-any** *class-map-name*} | **websense** {*class-map-name* | **match-any** *class-map-name*}} **Example:** `Router(config)# class-map type urlfilter websense websense-param-map`	Configures the class map for the URL filter with the same type of parameter map and enters Qos class-map configuration mode.
Step 6	**exit** **Example:** `Router(config-cmap)# exit`	Exits QoS class-map configuration mode.
Step 7	**policy-map type inspect urlfilter** *policy-map-name* **Example:** `Router(config)# policy-map type inspect urlfilter websense-policy`	Configures the URL filter policy.
Step 8	**service-policy urlfilter** *policy-map-name* **Example:** `Router(config)# service-policy urlfilter websense-policy`	Applies the URL filter policy under the inspect class as the service policy.

Configuring an IMAP Firewall Policy

Use the following tasks to configure an IMAP firewall policy:

- Configuring an IMAP Class Map, page 35
- Configuring an IMAP Policy Map, page 36

Configuring an IMAP Class Map

Perform this task to configure an IMAP class map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**reset**] [**secure-login**] [**timeout** *seconds*]
4. **class-map type inspect imap** [**match-any**] *class-map-name*
5. **log**

6. **match invalid-command**
7. **match login clear-text**
8. **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 inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**reset**] [**secure-login**] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name mail-guard imap`	Defines a set of inspection rules.
Step 4	**class-map type inspect imap** [**match-any**] *class-map-name* **Example:** `Router(config)# class-map type inspect imap imap-class`	Creates a class map for the IMAP to enter the match criteria, and enters QoS class-map configuration mode.
Step 5	**log** **Example:** `Router(config-cmap)# log`	Generates a log of messages.
Step 6	**match invalid-command** **Example:** `Router(config-cmap)# match invalid-command`	(Optional) Locates invalid commands on an IMAP connection.
Step 7	**match login clear-text** **Example:** `Router(config-cmap)# match login clear-text`	(Optional) Locates nonsecure login when an IMAP server is used.
Step 8	**exit** **Example:** `Router(config-cmap)# exit`	Exits QoS class-map configuration mode and returns to global configuration mode.

Configuring an IMAP Policy Map

Use this task to configure an IMAP policy map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect imap** *policy-map-name*
4. **class-type inspect imap** *imap-class-name*
5. **log**
6. **reset**

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	**policy-map type inspect imap** *policy-map-name* **Example:** `Router(config)# policy-map type inspect imap myimap-policy`	Creates a Layer 3 IMAP policy map and enters QoS policy-map configuration mode.
Step 4	**class-type inspect imap** *imap-class-name* **Example:** `Router(config-pmap)# class-type inspect imap pimap`	Creates a class map for the IMAP protocol.
Step 5	**log** **Example:** `Router(config-pmap)# log`	Generates a log (messages).
Step 6	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Resets a TCP connection if the data length of the SMTP body exceeds the value that you configured in the **class-map type inspect smtp** command.

Configuring an Instant Messenger Policy

Use the following tasks to configure an IM policy:

- Configuring an IM Class Map, page 38
- Configuring an IM Policy Map, page 38

You can create an IM policy for the following IM applications: America Online (AOL), ICQ, MSN Messenger, Yahoo Messenger, and Windows Messenger.

Configuring an IM Class Map

Use this task to configure a class map for any supported IM application.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class map type inspect** {**aol** | **msnmsgr** | **ymsgr** | **icq** | **winmsgr**} [**match-any**] *class-map-name*
4. **match service** {**any** | **text-chat**}

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	**class map type inspect** {**aol** | **msnmsgr** | **ymsgr** | **icq** | **winmsgr**} [**match-any**] *class-map-name* **Example:** `Router(config)# class map type inspect aol myaolclassmap`	Creates an IM type class map so you can begin adding match criteria and enters QoS class-map configuration mode.
Step 4	**match service** {**any** | **text-chat**} **Example:** `Router(config-cmap)# match service text-chat`	(Optional) Creates a match criterion on the basis of text chat messages (**text-chat**) or for any available service within a given IM protocol (**any**).

Configuring an IM Policy Map

Use this task to configure a policy map for any supported IM application.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy map type inspect** *protocol-name policy-map-name*
4. **class type inspect** {**aol** | **msnmsgr** | **ymsgr** | **icq** | **winmsgr**} *class-map-name*
5. **reset**

6. **log**
7. **allow**

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	**policy map type inspect** *protocol-name policy-map-name* **Example:** `Router(config)# policy map type inspect aol myaolpolicymap`	Creates an IM policy map and enters QoS policy-map configuration mode.
Step 4	**class type inspect** {**aol** | **msnmsgr** | **ymsgr** | **icq** | **winmsgr**} *class-map-name* **Example:** `Router(config-pmap)# class type inspect aol myaolclassmap`	Specifies a traffic class on which an action is to be performed. • *class-map-name*—This class map name should match the class map specified through the **class-map type inspect** command.
Step 5	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Resets the connection.
Step 6	**log** **Example:** `Router(config-pmap)# log`	(Optional) Generates a log message for the matched parameters.
Step 7	**allow** **Example:** `Router(config-pmap)# allow`	(Optional) Allows the connection.

What to Do Next

If you have not done so already, you must configure an IM-specific parameter map as shown in the task "Configuring a Layer 7 Protocol-Specific Parameter Map."

Configuring a Peer-to-Peer Policy

Use the following tasks to configure a P2P firewall policy:

You can create a P2P policy for the following P2P applications: eDonkey, FastTrack, Gnutella, and Kazaa Version 2.

Configuring a P2P Class Map

Use this task to configure a class map for any supported P2P application.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class map type inspect** {**edonkey** | **fasttrack** | **gnutella** | **kazaa2**} [**match-any**] *class-map-name*
4. **match file-transfer** [*regular-expression*]
5. **match search-file-name** [*regular-expression*]
6. **match text-chat** [*regular-expression*]

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	**class map type inspect** {**edonkey** | **fasttrack** | **gnutella** | **kazaa2**} [**match-any**] *class-map-name* **Example:** `Router(config)# class map type inspect edonkey myclassmap`	Creates a P2P type class map so you can begin adding match criteria and enters QoS class-map configuration mode.
Step 4	**match file-transfer** [*regular-expression*] **Example:** `Router(config-cmap)# match file-transfer *`	(Optional) Matches file transfer connections within any supported P2P protocol. **Note** To specify that all file transfer connections be identified by the traffic class, use "*" as the regular expression.

	Command or Action	Purpose
Step 5	**match search-file-name** [*regular-expression*] **Example:** `Router(config-cmap)# match search-file-name`	(Optional) Blocks filenames within a search request for clients using the eDonkey P2P application. **Note** This command is available only for the eDonkey P2P application.
Step 6	**match text-chat** [*regular-expression*] **Example:** `Router(config-cmap)# match text-chat`	(Optional) Blocks text chat messages between clients using the eDonkey P2P application. **Note** This command is available only for the eDonkey P2P application.

Configuring a P2P Policy Map

Use this task to configure a policy map for any supported P2P application.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy map type inspect p2p** *policy-map-name*
4. **class type inspect** {**edonkey** | **fasttrack** | **gnutella** | **kazaa2**} *class-map-name*
5. **reset**
6. **log**
7. **allow**

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	**policy map type inspect p2p** *policy-map-name* **Example:** `Router(config)# policy map type inspect p2p mypolicymap`	Creates a P2P policy map and enters QoS policy-map configuration mode.
Step 4	**class type inspect** {**edonkey** | **fasttrack** | **gnutella** | **kazaa2**} *class-map-name* **Example:** `Router(config-pmap)# class type inspect edonkey myclassmap`	Specifies a traffic class on which an action is to be performed and enters policy map configuration mode. • *class-map-name*—This class map name should match the class map specified through the **class-map type inspect** command.

	Command or Action	Purpose
Step 5	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Resets the connection.
Step 6	**log** **Example:** `Router(config-pmap)# log`	(Optional) Generates a log message for the matched parameters.
Step 7	**allow** **Example:** `Router(config-pmap)# allow`	(Optional) Allows the connection.

Configuring a POP3 Firewall Policy

Use the following tasks to configure a POP3 firewall policy:

- Configuring a POP3 Firewall Class Map, page 42
- Configuring a POP3 Firewall Policy Map, page 43

Configuring a POP3 Firewall Class Map

Use this task to configure a POP3 firewall class map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**reset**] [**secure-login**] [**timeout** *seconds*]
4. **class-map type inspect pop3** [**match-any**] *class-map-name*
5. **match invalid-command**
6. **match login clear-text**

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 inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**reset**] [**secure-login**] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name mail-guard pop3`	Defines a set of inspection rules.
Step 4	**class-map type inspect pop3** [**match-any**] *class-map-name* **Example:** `Router(config)# class-map type inspect pop3 pop3-class`	Creates a class map for the POP3 protocol so that you can enter match criteria and enters QoS class-map configuration mode.
Step 5	**match invalid-command** **Example:** `Router(config-cmap)# match invalid-command`	(Optional) Locates invalid commands on a POP3 server.
Step 6	**match login clear-text** **Example:** `Router(config-cmap)# match login clear-text`	(Optional) Finds a nonsecure login when using a POP3 server.

Configuring a POP3 Firewall Policy Map

Use this task to configure a POP3 firewall policy map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect pop3** *policy-map-name*
4. **class-type inspect pop3** *pop3-class-name*
5. **log**
6. **reset**

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	**policy-map type inspect pop3** *policy-map-name* **Example:** `Router(config)# policy-map type inspect pop3 mypop3-policy`	Creates a Layer 7 POP3 policy map and enters QoS policy-map configuration mode.
Step 4	**class-type inspect pop3** *pop3-class-name* **Example:** `Router(config-pmap)# class-type inspect pop3 pc1`	Creates a class map for the POP3 protocol.
Step 5	**log** **Example:** `Router(config-pmap)# log`	Generates a log (messages).
Step 6	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Resets a TCP connection if the data length of the SMTP body exceeds the value that you configured in the **class-map type inspect smtp** command.

Configuring an SMTP Firewall Policy

Use these tasks to configure an SMTP firewall policy:

- Configuring an SMTP Firewall Class Map, page 44
- Configuring an SMTP Firewall Policy Map, page 45

Configuring an SMTP Firewall Class Map

Use this task to configure an SMTP firewall class map.

Note To enable inspection for extended SMTP (ESMTP) in a class map, use the **match protocol smtp extended** command. See the "Restrictions for Zone-Based Policy Firewall" section on page 2 for more information on using this command in Cisco IOS Release 12.4(15)T.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** [**match-all** | **match-any**] *class-map-name*
4. **match data-length gt** *max-data-value*

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	**class-map type inspect smtp** [**match-all** | **match-any**] *class-map-name* **Example:** `Router(config)# class-map type inspect smtp smtp-class`	Creates a class map for the SMTP protocol so that you can enter match criteria and enters QoS class- map configuration mode.
Step 4	**match data-length gt** *max-data-value* **Example:** `Router(config-cmap)# match data-length gt 200000`	Determines if the amount of data transferred in an SMTP connection is above the configured limit.

Configuring an SMTP Firewall Policy Map

Use this task to configure an SMTP firewall policy map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect smtp** *policy-map-name*
4. **class-type inspect smtp** *smtp-class-name*
5. **reset**

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	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp mysymtp-policy`	Creates a Layer 7 SMTP policy map and enters QoS policy-map configuration mode.
Step 4	**class-type inspect smtp** *smtp-class-name* **Example:** `Router(config-pmap)# class-type inspect smtp sc`	Configures inspection parameters for the SMTP protocol.
Step 5	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Resets the TCP connection if the data length of the SMTP body exceeds the value that you configured in the **class-map type inspect smtp** command.

Configuring a SUNRPC Firewall Policy

Use these tasks to configure a SUNRPC firewall policy:

- Configuring a SUNRPC Firewall Class Map, page 46
- Configuring a SUNRPC Firewall Policy Map, page 47

Note If you are inspecting an RPC protocol (that is, you specified the **match protocol sunrpc** command in the Layer 4 class map), the Layer 7 SUNRPC policy map is required.

Configuring a SUNRPC Firewall Class Map

Use this task to configure a SUNRPC firewall class map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect sunrpc** [**match-any**] *class-map-name*
4. **match program-number** *program-number*

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	**class-map type inspect sunrpc** [**match-any**] *class-map-name* **Example:** `Router(config)# class-map type inspect sunrpc long-urls`	Creates a class map for the SUNRPC protocol so that you can enter match criteria and enters QoS class-map configuration mode.
Step 4	**match program-number** *program-number* **Example:** `Router(config-cmap)# match program-number 2345`	(Optional) Specifies the allowed Remote Procedure Call (RPC) protocol program number as a match criterion.

Configuring a SUNRPC Firewall Policy Map

Use this task to configure a SUNRPC firewall policy map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect sunrpc** *policy-map-name*
4. **class-type inspect sunrpc** *sunrpc-class-name*
5. **allow** [**wait-time** *minutes*]

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	**policy-map type inspect sunrpc** *policy-map-name* **Example:** `Router(config)# policy-map type inspect sunrpc my-rpc-policy`	Creates a Layer 7 SUNRPC policy map and enters policy-map configuration mode.
Step 4	**class-type inspect sunrpc** *sunrpc-class-name* **Example:** `Router(config-pmap)# class-type inspect sunrpc cs1`	Configures inspection parameters for the SUNRPC protocol.
Step 5	**allow** [**wait-time** *minutes*] **Example:** `Router(config-pmap)# allow wait-time 10`	(Optional) Allows the configured program number. • Specifies the number of minutes to keep a small hole in the firewall to allow subsequent connections from the same source address and to the same destination address and port. The default wait time is zero minutes. This keyword is available only for the RPC protocol.

Configuring an MSRPC Firewall Policy

Note If you are inspecting an RPC protocol (that is, you specified the **match protocol msrpc** command in the Layer 4 class map), the Layer 7 Microsoft Remote Procedure Call (MSRPC) policy map is required.

Perform this task to configure an MSRPC firewall policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type protocol-info msrpc** *parameter-map-name*
4. **timeout** *seconds*
5. **exit**
6. **class-map type inspect match-any** *class-map-name*
7. **match protocol msrpc**
8. **match protocol msrpc-smb-netbios**
9. **exit**
10. **policy-map type inspect** *policy-map-name*
11. **class type inspect** *class-map-name*
12. **inspect**
13. **exit**
14. **class class-default**
15. **drop**

16. **exit**
17. **exit**
18. **zone security** *security-zone-name*
19. **exit**
20. **zone security** *security-zone-name*
21. **exit**
22. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
23. **service-policy type inspect** *policy-map-name*
24. **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	**parameter-map type protocol-info msrpc** *parameter-map-name* **Example:** `Router(config)# parameter-map type protocol-info msrpc para-map`	Defines an application-specific parameter map and enters profile configuration mode.
Step 4	**timeout** *seconds* **Example:** `Router(config-profile)# timeout 60`	Configures the MSRPC endpoint mapper (EPM) timeout.
Step 5	**exit** **Example:** `Router(config-profile)# exit`	Exits profile configuration mode and enters global configuration mode.
Step 6	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any c-map`	Creates an inspect type class map for the traffic class and enters QoS class-map configuration mode.
Step 7	**match protocol msrpc** **Example:** `Router(config-cmap)# match protocol msrpc`	Configures the match criteria for a class map on the basis of a specified protocol. • Only Cisco IOS stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.

	Command or Action	Purpose
Step 8	**match protocol msrpc-smb-netbios** **Example:** `Router(config-cmap)# match protocol msrpc-smb-netbios`	Configures the match criteria for a class map on the basis of a specified protocol. • Only Cisco IOS stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 9	**exit** **Example:** `Router(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 10	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect p-map`	Creates a Layer 3 and Layer 4 inspect type policy map and enters QoS policy-map configuration mode.
Step 11	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect c-map`	Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 12	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS stateful packet inspection.
Step 13	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 14	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies the matching of the system default class and enters QoS policy-map class configuration mode. • If the system default class is not specified, then unclassified packets are matched.
Step 15	**drop** **Example:** `Router(config-pmap-c)# drop`	Drops packets that are matched with the defined class.
Step 16	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 17	**exit** **Example:** `Router(config-pmap)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 18	**zone security** *security-zone-name* **Example:** `Router(config)# zone security in-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 19	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 20	**zone security** *security-zone-name* **Example:** `Router(config)# zone security out-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 21	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 22	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Router(config)# zone-pair security in-out source in-zone destination out-zone`	Creates a zone pair and enters security zone configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 23	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone)# service-policy type inspect p-map`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 24	**end** **Example:** `Router(config-sec-zone)# end`	Exits security zone configuration mode and enters privileged EXEC mode.

Creating Security Zones and Zone Pairs, and Attaching a Policy Map to a Zone Pair

You need two security zones to create a zone pair. However, you can create only one security zone and use a system-defined security zone called “self.” Note that if you select a self zone, you cannot configure inspect policing.

Use this process to complete the following tasks:

- Create at least one security zone
- Define zone pairs
- Assign interfaces to security zones
- Attach a policy map to a zone pair

Tip Before you create zones, think about what should constitute the zones. The general guideline is that you should group interfaces that are similar when they are viewed from a security perspective.

Security Zone Restrictions

- An interface cannot be part of a zone and legacy inspect policy at the same time.
- An interface can be a member of only one security zone.
- When an interface is a member of a security zone, all traffic to and from that interface is blocked unless you configure an explicit interzone policy on a zone pair involving that zone.
- Traffic cannot flow between an interface that is a member of a security zone and an interface that is not a member of a security zone because a policy can be applied only between two zones.
- For traffic to flow among all the interfaces in a router, all the interfaces must be members of one security zone or another. This is particularly important because after you make an interface a member of a security zone, a policy action (such as **inspect** or **pass**) must explicitly allow packets. Otherwise, packets are dropped.
- If an interface on a router cannot be part of a security zone or firewall policy, you may have to put that interface in a security zone and configure a "pass all" policy (that is, a "dummy" policy) between that zone and other zones to which a traffic flow is desired.
- You cannot apply an access control list (ACL) between security zones or on a zone pair.
- An ACL cannot be applied between security zones and zone pairs. Include the ACL configuration in a class map, and use policy maps to drop traffic.
- An ACL on an interface that is a zone member should not be restrictive (strict).
- All interfaces in a security zone must belong to the same virtual routing and forwarding (VRF) instance.
- You can configure policies between security zones whose member interfaces are in separate VRFs. However, traffic may not flow between these VRFs if the configuration does not allow it.
- If traffic does not flow between VRFs (because route-leaking between VRFs is not configured), the policy across the VRFs is not executed. This is a misconfiguration on the routing side, not on the policy side.
- Traffic between interfaces in the same security zone is not subjected to any policy; the traffic passes freely.
- The source and the destination zones in a zone pair must be the type security.
- The same zone cannot be defined as both the source and the destination.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **description** *line-of-description*
5. **exit**
6. **zone-pair security** *zone-pair-name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]

7. **description** *line-of-description*
8. **exit**
9. **interface** *type number*
10. **zone-member security** *zone-name*
11. **exit**
12. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**} **destination** [**self** | *destination-zone-name*]
13. **service-policy type inspect** *policy-map-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	**zone security** *zone-name* **Example:** `Router(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**description** *line-of-description* **Example:** `Router(config-sec-zone)# description Internet Traffic`	(Optional) Describes the zone.
Step 5	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 6	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 7	**description** *line-of-description* **Example:** `Router(config-sec-zone)# description accounting network`	(Optional) Describes the zone pair.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface for configuration and enters interface configuration mode.
Step 10	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 12	**zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**]} **destination** [**self** | *destination-zone-name*] **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone pair configuration mode.
Step 13	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect p2`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

Configuring the Cisco IOS Firewall with WAAS

Use the task in this section to enable Cisco IOS firewall inspection so that WAAS optimization can be discovered.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp** *service-id*
4. **ip inspect waas enable**

5. **class-map type inspect** *class-name*
6. **match protocol** *protocol-name* [**signature**]
7. **exit**
8. **policy-map type inspect** *policy-map-name*
9. **class class-default**
10. **class-map type inspect** *class-name*
11. **inspect**
12. **exit**
13. **exit**
14. **zone security** *zone-name*
15. **description** *line-of-description*
16. **exit**
17. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
18. **description** *line-of-description*
19. **exit**
20. **interface** *type number*
21. **description** *line-of-description*
22. **zone-member security** *zone-name*
23. **ip address** *ip-address*
24. **ip wccp** {*service-id* {**group-listen** | **redirect** {**in** | **out**}} | **redirect exclude in** | **web-cache** {**group-listen** | **redirect** {**in** | **out**}}
25. **exit**
26. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**} **destination** [**self** | *destination-zone-name*]
27. **service-policy type inspect** *policy-map-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.

	Command or Action	Purpose
Step 3	**ip wccp** *service-id* **Example:** `Router(config)# ip wccp 61`	Enters the WCCP dynamically defined service identifier number.
Step 4	**ip inspect waas enable** **Example:** `Router(config)# ip inspect WAAS enable`	Enables the Cisco IOS firewall inspection so that WAAS optimization can be discovered. **Note** If an ISR router with Cisco IOS Firewall is deployed as an intermediary router inside the WAAS optimization path, the **ip inspect waas enable** command needs to be used to enable WAAS awareness and interoperability. If the router were not configured for optimization awareness, optimized traffic would violate TCP activity expectations, and the firewall would drop the traffic.
Step 5	**class-map type inspect** *class-name* **Example:** `Router(config)# class-map type inspect most-traffic`	Creates an inspect type class map for the traffic class and enters QoS class-map configuration mode. **Note** The **class-map type inspect most-traffic** command is hidden.
Step 6	**match protocol** *protocol-name* [**signature**] **Example:** `Router(config-cmap)# match protocol http`	Configures the match criteria for a class map on the basis of a specified protocol and enters security zone configuration mode. • Only Cisco IOS stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps. • **signature**—Signature-based classification for peer-to-peer (P2P) packets is enabled.
Step 7	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 8	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect p1`	Creates a Layer 3 and Layer 4 inspect type policy map and enters QoS policy-map configuration mode.
Step 9	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies the matching of the system default class. • If the system default class is not to be specified, then unclassified packets are matched.
Step 10	**class-map type inspect** *class-name* **Example:** `Router(config-pmap)# class-map type inspect most-traffic`	Specifies the firewall traffic (class) map on which an action is to be performed.

	Command or Action	Purpose
Step 11	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS stateful packet inspection.
Step 12	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy map configuration mode.
Step 13	**exit** **Example:** `Router(config-pmap)# exit`	Returns to global configuration mode.
Step 14	**zone security** *zone-name* **Example:** `Router(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 15	**description** *line-of-description* **Example:** `Router(config-sec-zone)# description Internet Traffic`	(Optional) Describes the zone.
Step 16	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 17	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 18	**description** *line-of-description* **Example:** `Router(config-sec-zone)# description accounting network`	(Optional) Describes the zone pair.
Step 19	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 20	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 21	**description** *line-of-description* **Example:** `Router(config-if)# description zone interface`	(Optional) Describes the interface.
Step 22	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 23	**ip address** *ip-address* **Example:** `Router(config-if)# ip address 10.70.0.1 255.255.255.0`	Assigns the interface IP address for the security zone and enters interface configuration mode.
Step 24	**ip wccp** {*service-id* {**group-listen** | **redirect** {**in** | **out**}} | **redirect exclude in** | **web-cache** {**group-listen** | **redirect** {**in** | **out**}} **Example:** `Router(config-if)# ip wccp 61 redirect in`	Specifies the following WCCP parameters on the interface: • The *service-id* argument defines a service identifier number from 1 to 254. • The **redirect exclude in** keywords are used to exclude inbound packets from outbound redirection. • The **web-cache** keyword is used to define the standard web caching service. • The **group-listen** keyword is used for discovering multicasted WCCP protocol packets. • The **in** keyword is used to redirect to a cache engine the appropriate inbound packets. • The **out** keyword is used to redirect to a cache engine the appropriate outbound packets.
Step 25	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 26	**zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**]} **destination** [**self** | *destination-zone-name*] **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone pair configuration mode.
Step 27	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect p2`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

Configuration Examples for Zone-Based Policy Firewall

Example: Configuring Layer 3 and Layer 4 Firewall Policies: Example

The following example shows a Layer 3 or Layer 4 top-level policy. Traffic is matched to access control list 199. There is deep-packet HTTP inspection. Configuring the **match access-group** 101 filter enables Layer 4 inspection. As a result, Layer 7 inspection is omitted unless the class-map is of type **mach-all**.

```
class-map type inspect match-all http-traffic
 match protocol http
 match access-group 101
policy-map type inspect mypolicy
 class type inspect http-traffic
  inspect
  service-policy http http-policy
```

Example: Configuring Layer 7 Firewall Policies

The following example shows how to match HTTP sessions that have a URL length greater than 500. The Layer 7 policy action is **reset**.

```
class-map type inspect http long-urls
 match request uri length gt 500
policy-map type inspect http http-policy
 class type inspect http long-urls
```

```
  reset
```

The following example shows how to enable inspection for ESMTP by including the **extended** keyword:

```
class-map type inspect c1
 match protocol smtp extended

policy-map type inspect p1
 class type inspect c1
  inspect
```

Now the **service-policy type inspect smtp** command is optional and can be entered after the **inspect** command.

Example: Configuring a Security Zone:

The following example shows how to create security zone z1, which is called Internet Traffic:

```
zone security z1
 description Internet Traffic
```

Example: Configuring a Zone Pair

A zone-based firewall drops a packet if it is not explicitly allowed by a rule or policy in contrast to a legacy firewall, which permits a packet if it is not explicitly denied by a rule or policy by default.

A zone-based firewall behaves differently in handling intermittent ICMP responses generated within a zone as a result of the traffic flowing between in-zones and out-zones.

In a configuration where an explicit policy is configured for the self zone to go out of its zone and for the traffic moving between the in-zone and out-zone, if any intermittent ICMP responses are generated, then the zone-based firewall looks for a explicit permit rule for the ICMP protocol in the self zone to go out of its zone. An explicit inspect rule for the ICMP protocol for the self zone to go out-zone may not help because there is not a session associated with the intermittent ICMP responses.

The following example shows how to create zones z1 and z2, describes the zones, and specifies that the firewall policy map is applied in zone z2 for traffic flowing between the zones:

```
zone security z1
 description finance department networks

zone security z2
 description engineering services network

zone-pair security zp source z1 destination z2
```

Example: Assigning an Interface to a Security Zone

The following example shows how to attach Ethernet interface 0 to zone z1:

```
interface ethernet0
 zone-member security z1
```

Example: Attaching a Policy Map to a Zone Pair

The following example shows how to attach a firewall policy map to the target zone pair p1:

```
zone-pair security zp source z1 destination z2
 service-policy type inspect p1
```

Example: Configuring a URL Filter Policy: Websense

The following example shows how to configure a URL filter policy for Websense:

Example: Websense Server Configuration

The following example shows how to configure the Websense server:

```
parameter-map type urlfpolicy websense websense-param-map
 server fw21-ss1-bldr.example.com timeout 30
 source-interface Loopback0
 truncate script-parameters
 cache-size maximum-entries 100
 cache-entry-lifetime 1
 block-page redirect-url http://abc.example.com
```

Example: Configuring the Websense Class Map

The following example shows how to configure the Websense class map:

```
class-map type urlfilter websense match-any websense-class
 match server-response any
```

Example: Configuring the Websense URL Filter Policy

The following example shows how to configure the Websense URL filter policy:

```
policy-map type inspect urlfilter websense-policy
 parameter type urlfpolicy websense websense-param-map
 class type urlfilter websense websense-class
  server-specified-action
  log
```

Example: Applying the URL filter to Firewall Policy

The following example shows how to apply the URL filter to the firewall policy.:

```
policy-map type inspect websense-global-policy
 class type inspect http-class
  inspect global
  service-policy urlfilter websense-policy
```

Example: Cisco IOS Firewall Configuration with WAAS

The following example provides an end-to-end WAAS traffic flow optimization configuration for the Cisco IOS firewall that uses WCCP to redirect traffic to a WAE device for traffic interception.

The following configuration example prevents traffic from being dropped between security zone members because the integrated-service-engine interface is configured on a different zone and each security zone member is assigned an interface. This change was made to the Cisco IOS firewall configuration in Cisco IOS Release 12.4(20)T and 12.4(22)T to address the different input interfaces.

```
ip wccp 61
ip wccp 62
ip inspect waas enable
class-map type inspect most-traffic
match protocol icmp
match protocol ftp
match protocol tcp
match protocol udp
policy—map type inspect p1
class type inspect most—traffic
inspect
class class—default
zone security zone-hr
zone security zone-outside
zone security z-waas
zone—pair security hr—out source zone-hr destination zone-outside
service—policy type inspect p1
zone—pair security out—hr source zone-outside destination zone-hr
service—policy type inspect p1
zone—pair security eng—out source zone-eng destination zone-outside
service—policy type inspect p1

interface GigabitEthernet0/0
description Trusted interface
ipaddress 10.70.0.1 255.255.255.0
ip wccp 61 redirect in
zone—member security zone-hr

interface GigabitEthernet0/0
description Trusted interface
ipaddress 10.71.0.2 255.255.255.0
ip wccp 61 redirect in
zone—member security zone-eng

interface GigabitEthernet0/1
description Untrusted interface
ipaddress 10.72.2.3 255.255.255.0
ip wccp 62 redirect in
zone—member security zone-outside
```

Note The new configuration in Cisco IOS Release 12.4(20)T and 12.4(22)T places the integrated service engine in its own zone and need not be part of any zone pair. The zone pairs are configured between zone-hr (zone-out) and zone-eng (zone-output).

```
interface Integrated—Service—Enginel/0
ipaddress 10.70.100.1 255.255.255.252
ip wccp redirect exclude in
zone—member security z-waas
```

Example: Protocol Match Data Not Incrementing for a Class Map

The following configuration example causes the match counter problem in the **show policy-map type inspect zone-pair** command output:

```
class-map type inspect match-any y
 match protocol tcp
 match protocol icmp
class-map type inspect match-all x
 match class y
```

However, cumulative counters for the configuration is displayed in the **show policy-map type inspect zone-pair command** output if the class map matches any class map:

```
show policy-map type inspect zone session

policy exists on zp zp
 Zone-pair: zp

  Service-policy inspect : fw

    Class-map: x (match-any)
      Match: class-map match-any y
        2 packets, 48 bytes   <======= Cumulative class map counters are incrementing.
        30 second rate 0 bps
        Match: protocol tcp
          0 packets, 0 bytes     <==== The match for the protocol is not incrementing.
          30 second rate 0 bps
        Match: protocol icmp
          0 packets, 0 bytes
          30 second rate 0 bps

   Inspect

      Number of Established Sessions = 1
      Established Sessions
        Session 53105C0 (10.1.1.2:19180)=>(172.1.1.2:23) telnet:tcp SIS_OPEN
          Created 00:00:02, Last heard 00:00:02
          Bytes sent (initiator:responder) [30:69]

    Class-map: class-default (match-any)
      Match: any
      Drop
        0 packets, 0 bytes
```

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*
Quality of service commands	*Cisco IOS Quality of Service Solutions Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	—

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 Zone-Based Policy Firewall

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

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

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

Table 1 Feature Information for Zone-Based Policy Firewall

Feature Name	Releases	Feature Information
Application Inspection And Control for HTTP—Phase 2	12.4(9)T	This feature extends support for HTTP application firewall policies. The following section provides information about this feature: • Configuring an HTTP Firewall Policy, page 29 The following commands were introduced or modified by this feature: **match body regex**, **match header count**, **match header length**, **match header regex**, **match request length**, **match request regex**, **match response status-line regex**.
E-mail Inspection Engine	15.1(1)S	This feature allows the users to inspect POP3, IMAP, and E/SMTP e-mail traffic contained in SSL VPN tunneled connections that traverse the Cisco IOS router. The following section provides information about this feature: • Configuring an IMAP Class Map, page 35

Table 1 Feature Information for Zone-Based Policy Firewall (continued)

Feature Name	Releases	Feature Information
Rate-limiting Inspected Traffic	12.4(9)T	This feature allows users to rate limit traffic within a Cisco IOS firewall (inspect) policy. Also, users can limit the absolute number of sessions that can exist on a zone pair. The following sections provide information about this feature: • Rate Limiting (Policing) Traffic Within a Layer 3 and Layer 4 Policy Map, page 9 • Creating an Inspect Parameter Map, page 20 The following commands were introduced by this feature: **police** (zone policy), **sessions maximum**.
P2P Application Inspection and Control—Phase 1	12.4(9)T 12.4(20)T	This feature introduces support for identifying and enforcing a configured policy for the following peer-to-peer applications: eDonkey, FastTrack, Gnutella Version 2, and Kazaa Version 2. Support for identifying and enforcing a configured policy for the following Instant Messenger applications is also introduced: AOL, MSN Messenger and Yahoo Messenger. In Release 12.4(20)T, support was added for the following applications: H.323 VoIP and SIP. In Release 12.4(20)T, support for the following IM applications was also added: ICQ and Windows Messenger. The following sections provide information about this feature: • Configuring a Layer 7 Protocol-Specific Parameter Map, page 24 • Configuring an Instant Messenger Policy, page 37 • Configuring a Peer-to-Peer Policy, page 39 The following commands were introduced or modified by this feature: **class-map type inspect**, **class type inspect**, **clear parameter-map type protocol-info**, **debug policy-firewall, match file-transfer**, **match protocol** (zone), **match search-file-name**, **match service**, **match text-chat**, **parameter-map type**, **policy-map type inspect, server** (parameter-map), **show parameter-map type protocol-info**.
Zone-Based Firewall Support for MSRPC	15.1(4)M	This feature introduces zone-based policy firewall support for Microsoft Remote Procedure Calls. The following section provides information about this feature: • Configuring an MSRPC Firewall Policy, page 48

Table 1 Feature Information for Zone-Based Policy Firewall (continued)

Feature Name	Releases	Feature Information
Zone-Based Policy Firewall	12.4(6)T	This feature provides a Cisco IOS unidirectional firewall policy between groups of interfaces known as zones. The following commands were introduced or modified by this feature: **class-map type inspect**, **class type inspect**, **clear parameter-map type protocol-info**, **debug policy-firewall**, **match body regex**, **match file-transfer**, **match header count**, **match header length**, **match header regex**, **match protocol** (**zone**), **match request length**, **match request regex**, **match response status-line regex**, **match search-file-name**, **match service**, **match text-chat**, **parameter-map type**, **policy-map type inspect**, **server** (parameter-map), **service-policy** (policy-map), **service-policy type inspect**, **show parameter-map type protocol-info**.
Zone Based Firewall (ZBFW) Usability and Manageability Features	15.0(1)M	The ZBFW usability and manageability features covered in this document are OoO packet processing support in zone based firewalls, intrazone support in zone-based firewalls and enhanced debug capabilities. The following sections provide information about this feature: • Out-of-Order Packet Processing Support in the Zone-Based Firewall Application, page 14 • Intrazone Support in the Zone-Based Firewall Application, page 15 • Configuring OoO Packet Processing Support in the Zone-Based Firewall Applications, page 25 • Configuring Intrazone Support in the Zone-Based Firewall Applications, page 27 The following commands were introduced or modified by this feature: **clear ip ips statistics**, **debug cce dp named-db inspect**, **debug policy-firewall**, **debug ip virtual-reassembly list**, **parameter-map type ooo global**, **show parameter-map type ooo global**, **zone-pair security**. In Cisco IOS Release 15.1(1)T, the following commands were introduced or modified: **class-map type inspect**, **clear policy-firewall**, **log** (parameter-map type), **match request regex**, **parameter-map type inspect**, **show parameter-map type inspect**, **show policy-firewall config**, **show policy-firewall mib**, **show policy-firewall sessions**, **show policy-firewall stats**, **show policy-firewall summary-log**.