Security Configuration Guide: Zone-Based Policy Firewall, Cisco IOS Release 15M&T

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

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

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2017 Cisco Systems, Inc. All rights reserved.

CONTENTS

CHAPTER 1

Zone-Based Policy Firewalls

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

Note Cisco IOS XE supports Virtual Fragmentation Reassembly (VFR) on zone-based firewall configuration. When you enable the firewall on an interface by adding the interface to a zone, VFR is configured automatically on the same interface.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Zone-Based Policy Firewall

- Before you create zones, you must consider what should constitute 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 Zone-Based Policy Firewall feature in Cisco IOS Release 12.4(11)T2 and later releases.

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 recommended.
- In Cisco IOS Releases 12.4(20)T and 12.4(15)T, 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. The problem with counters exists regardless of whether the top-level class map uses the **match-any** or **match-all** keyword. For more information, see the "Example: Protocol Match Data Not Incrementing for a Class Map" section.
- In Cisco IOS Release 12.4(15)T, if the Simple Mail Transfer Protocol (SMTP) is configured and you need to configure the Extended SMTP (ESMTP), you must configure the **no match protocol smtp** command before configuring 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, the following error is displayed: `%Cannot add this filter. Remove match protocol smtp filter and then add this filter.`
- In a Wide-Area Application Services (WAAS) and firewall configuration, all packets processed by a Wide Area Application Engine (WAE) must pass through the firewall in both directions to support the Web Cache Coordination Protocol (WCCP). This situation occurs because the Layer 2 redirect is not available in Cisco IOS 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.
- In a WAAS and firewall configuration, WCCP does not support traffic redirection using policy-based routing (PBR).
- The zone-based firewall cannot interoperate with WAAS and WCCP, when WCCP is configured with Layer 2 redirect method. The firewall only supports generic routing encapsulation (GRE) redirection.
- The zone-based firewall does not support when Layer 2 redirect is configured as a redirection method in WAAS. Only GRE as a redirection method is supported.
- Stateful inspection support for multicast traffic is not supported between any zones, including the self zone. Use Control Plane Policing for the protection of the control plane against multicast traffic.
- 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).
- A UDP-based traceroute is not supported through Internet Control Message Protocol (ICMP) inspection.

- To allow GRE and Encapsulating Security Payload (ESP) protocol traffic through a zone-based policy firewall, use the **pass** command. The GRE and the ESP protocols do not support stateful inspection and if you use the **inspect** command, the traffic for these protocols is dropped.
- In Cisco IOS Release 15.3(1)T and later releases, the peer-to-peer protocols are deprecated. You cannot configure the peer-to-peer protocols with zone-based policy firewalls.
- The zone-based firewall supports only Skinny Client Control Protocol (SCCP) protocol versions up to 17. SCCP versions above 17 are not tested or supported. If you are using an SCCP version that is above 17, either use the **pass** command instead of the **inspect** command or allow the out-to-in traffic through access control lists (ACLs).
- Configuring zone-based policy firewall high availability with Network Address Translation (NAT) and NAT high availability with zone-based policy firewalls is not recommended.
- If you have configured multiple class matching for Layer 7 policies, the reset action takes precedence over other actions such as pass and allow. Unlike Layer 4 policies, the zone-based firewall classification runs through all class maps even though the traffic has already matched a class map.

Information About Zone-Based Policy Firewalls

Top-Level Class Maps and Policy Maps

Top-level class maps allow you to identify the traffic stream at a high level. Identifying the traffic stream is accomplished by using the **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 maps to a target (zone pair).

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

With CSCto44113 fix, only Layer 4 policy maps will be inspected by the firewall when you configure the **access-group match** command. Prior to this fix, when the **access-group match** command was configured, both Layer 4 and Layer 7 policy maps were inspected.

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 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. 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 to specify where a Cisco firewall is applied.

For example, on a device, 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 subject to any policy and 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 the 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 (except traffic going to the device or initiated by the device) between that interface and an interface within a different zone is dropped by default. To permit traffic to and from a zone-member interface and another interface, you must make that zone part of a zone pair and apply a policy to that zone pair. If the policy permits traffic through **inspect** or **pass** actions, traffic can flow through the interface.

The following are basic rules to consider when setting up zones:

- Traffic from a zone interface to a nonzone interface or from a nonzone interface to a zone interface is always dropped; unless default zones are enabled (default zone is a nonzone interface).
- 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.
- A zone pair can be configured with a zone as both source and 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.
- An interface cannot be part of a zone and a 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 interfaces in a device, all 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 device cannot be part of a security zone or firewall policy, you may have to add 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 VPN 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 VRFs is not executed. This is a configuration mistake on the routing side, not on the policy side.
- Traffic between interfaces in the same security zone is not subject to any policy; traffic passes freely.
- Source and destination zones in a zone pair must be of the type security.
- The same zone cannot be defined as both source and destination zones.

A policy is applied to an initiating packet of a traffic flow. After the initial packet has been classified and permitted, traffic flows between peers with no further reclassification of the packet (this means that bidirectional traffic flow is allowed after the initial classification). If you have a zone pair between Zone Z1 and Zone Z2, and no zone pair between Zone Z2 and Zone Z1, all traffic that is initiated from Zone Z2 is blocked. Traffic from Zone Z1 to Zone Z2 is permitted or denied based on the zone pair policy.

For traffic to flow among all interfaces in a device, all interfaces must be members of security zones or the default zone.

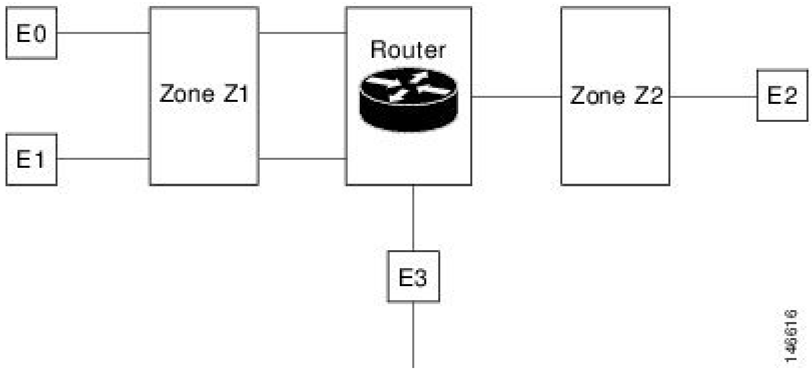
It is not necessary for all device interfaces to be members of security zones.

The figure below 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. If no policy is configured for Z1 and Z2, traffic will flow freely between E0 and E1, but not between E0 or E1 to E2. A zone pair and policy may be created to inspect this traffic.
- 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, or E2 unless default zones are enabled and a zone pair is created between the default zone and other zones.

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 a configuration common to specific users, plus device-dependent information. The template contains Cisco software interface commands that are applied to virtual access interfaces. To configure a virtual template interface, use the **interface virtual-template** command.

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

The **zone-member security** command adds the dynamic interface to 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 source and destination zones. The source and destination zones of a zone pair must be security zones.

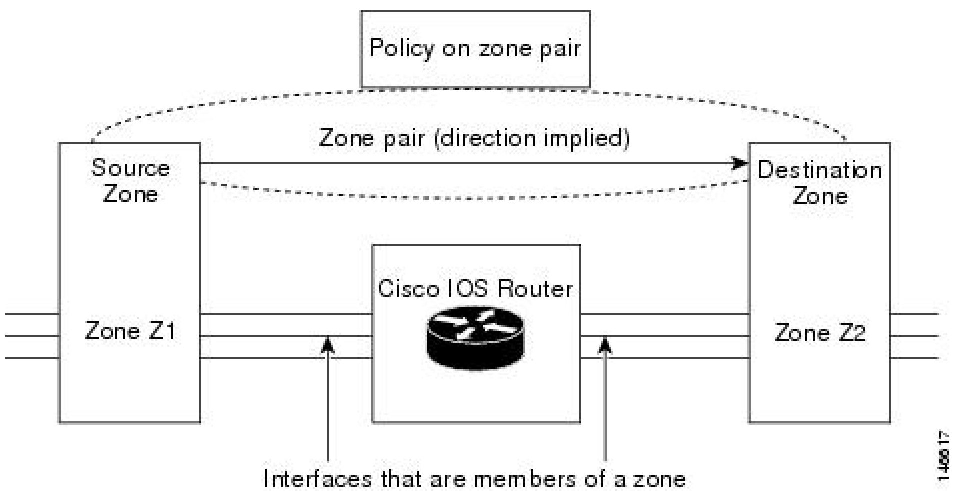
You can select the default or self zone as either the source or the destination zone. The self zone is a system-defined zone which does not have any interfaces as members. A zone pair that includes the self zone, along with the associated policy, applies to traffic directed to the device or traffic generated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

The most common usage of firewall is to apply them to traffic through a device, so you 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.

The figure below 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 zone pairs, traffic is dropped. However, it is not necessary to configure a zone pair and a service policy solely for the return traffic. By default, return traffic is not allowed. If a service policy inspects the traffic in the forward direction and there is no zone pair and service policy for the return traffic, the return traffic is inspected. If a service policy passes the traffic in the forward direction and there is no zone pair and service policy for the return traffic, the return traffic is dropped. In both these cases, you need to configure a zone pair and a service policy to allow the return traffic. In the above figure, it is not mandatory that you configure a zone pair source and destination for allowing return traffic from Z2 to Z1. The service policy on Z1 to Z2 zone pair takes care of it.

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 when handling intermittent Internet Control Message Protocol (ICMP) responses generated within a zone because 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 an explicit permit rule for the ICMP in the self zone to go out of its zone. An explicit inspect rule for the ICMP for the self zone to go out-zone may not help because there is no session associated with the intermittent ICMP responses.

Zones and Inspection

Zone-based policy firewalls examine 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 individual flows. Traffic with the inspect action will create a connection in the firewall table and be subject to state checking. Traffic with the pass action will bypass the zone firewall completely, not creating any sessions.

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

Zones and ACLs

Access control lists (ACLs) applied to interfaces that are members of zones are processed before the policy is applied on the zone pair. You must ensure that interface ACLs do not interfere with the policy firewall traffic when there are policies between zones.

Pinholes (ports opened through a firewall that allows applications-controlled access to a protected network) are not punched for return traffic in interface ACLs.

Zones and VRF-Aware Firewalls

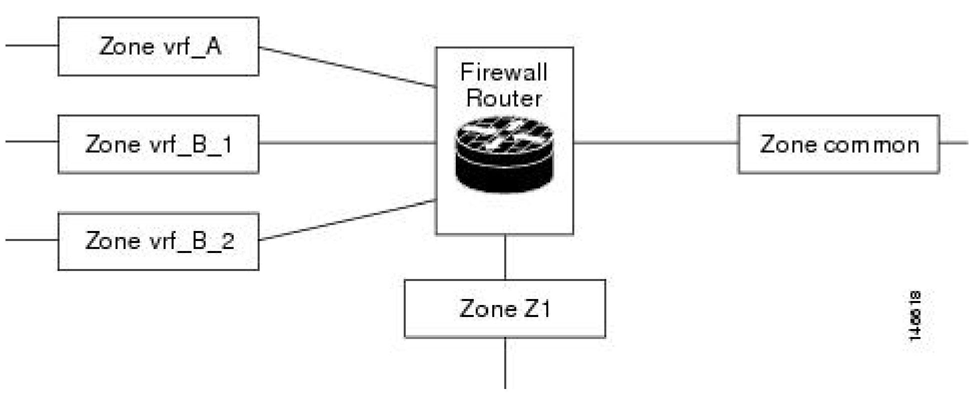
The Cisco firewall is VPN routing and forwarding (VRF)-aware. It handles IP address overlap across different VRFs, separate thresholds, and timeouts for VRFs. All interfaces in a zone must belong to the same VRF.

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 the figure below.

When multiple VRFs are configured on a device 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 the figure above, 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 make sure that traffic can flow between them.

You do not 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 firewall supports transparent firewalls where the interfaces are placed in bridging mode and the firewall inspects 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 zone member. 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

The Cisco 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, 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 that is typically associated with a traffic class. For example, **inspect**, **drop**, and **pass** are actions.

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

- Define a match criterion (class map).
- Associate actions to the match criterion (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), that are determined by how the **service-policy** command is configured, are checked against 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.

To log firewall drop messages, enable the **drop-log** command under the class-default class in the policy map. For example, consider the following policy map:

```
policy-map type inspect in-out-pol
 class type inspect in-out
  inspect
 class class-default
  drop-log
policy-map type inspect out-in-pol
 class type inspect out-in
  inspect
 class class-default
  drop-log
```

To log dropped packets for an inspect parameter map, use the **log dropped-packets enable** command. The following example shows how to configure logging of dropped packets due to an inspect policy:

```
parameter-map type inspect global
 log dropped-packets enable
```

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 are of type inspect and 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 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 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:

```
Device(config)# class-map type inspect match-all c1
Device(config-cmap)# match access-group 101
Device(config-cmap)# match protocol http
Device(config)# policy-map type inspect p1
Device(config-pmap)# class type inspect c1
Device(config-pmap-c)# drop
```

Class-Map Configuration Restriction

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

```
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 is handled by the service-specific capabilities of HTTP inspection. If the “match” lines are reversed, and the traffic encounters the **match protocol tcp** command before it is compared to the **match protocol http** command, the traffic will be classified as TCP traffic and inspected according to the capabilities of the TCP inspection component of the firewall. If match protocol TCP is configured first, it will create issues for services such as FTP and TFTP and for multimedia and voice signaling services such as H.323, Real Time Streaming Protocol (RTSP), Session Initiation Protocol (SIP), and Skinny. These services require additional inspection capabilities to recognize more complex activities.

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

Depending on your releases, you can use the **police** command within an inspect policy to limit the number of concurrent connections allowed for applications such as Instant Messenger (IM) and peer-to-peer (P2P).

To use the **police** command, you must enable Cisco stateful packet inspection within the inspect policy map. If you configure the **police** command without configuring 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 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 and not to an interface. The police action is enforced on traffic that traverses the zone pair. (The direction of the traffic is inherent to the specification of the zone pair.) Thus, a quality of service (QoS) policy that contains a police action can be present on interfaces that make up a zone pair and in an inspect policy map applied across the zone pair. If both police actions are configured, the zone pair police action is executed after the input interface police action, but before the output interface police action. There is no interaction between QoS and the inspect police actions.

Police Restrictions

- The police action is not allowed in policies that are attached to zone pairs that involves a “self” zone. Use Control Plane Policing 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). The DPI functionality is delivered through Layer 7 class maps and policy maps.

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

The type of class map (for example, HTTP) determines the match criteria that you can use. 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 of the same type.

To create a Layer 7 policy map, specify the protocol in the **policy-map type inspect** command. For example, to create a Layer 7 HTTP policy map, use the **policy-map type inspect http** *policy-map-name* command. Enter the name of the HTTP policy-map for the *policy-map-name* argument.

If you do not specify a protocol name (for example, if you use the **policy-map type inspect** command), you will create 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** command and specify the application name (that is, HTTP, Internet Message Access Protocol [IMAP], Post Office Protocol, version 3 [POP3], Simple Mail Transfer Protocol [SMTP], or SUN Remote Procedure Call [SUNRPC]). The parent class for a Layer 7 policy should have an explicit match criterion that matches 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.

If you have configured multiple classes matching for Layer 7 policies, the reset action takes precedence over other actions such as pass and allow. Unlike Layer 4 policies, the zone-based firewall classification runs through all class maps even though the traffic has already matched a class map.

In the following example, policy map p1 has two classes, c1 and c2 attached to it. However, if the traffic matches both c1 and c2, the reset action has precedence over the allow action.

```
Device(config)# policy-map type inspect p1
Device(config-pmap)# class type inspect c1
Device(config-pmap-c)# allow
!
Device(config-pmap)# class type inspect c2
Device(config-pmap-c)# reset
!
```

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 peer-to-peer protocol.
- FastTrack traffic peer-to-peer protocol.
- Gnutella Version 2 traffic peer-to-peer protocol.
- H.323 VoIP Protocol Version 4.
- HTTP—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 is shared.
- I Seek You (ICQ) IM protocol.
- Kazaa Version 2 peer-to-peer 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, on page 29" 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. The class-default class is always the last class in a policy map.

You can define explicit actions for a group of packets that does not match any of the user-defined classes. If you do not configure any actions for the class-default class 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.

```
Device(config)# class-map type inspect match-all c1
Device(config-cmap)# match protocol http
Device(config-cmap)# exit
Device(config)# policy-map type inspect p1
Device(config-pmap)# class type inspect c1
Device(config-pmap-c)# drop
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(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 that uses the preexisting 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 two 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, parameters in the lower levels override those in the top levels.

- Protocol-specific parameter map

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

Firewall and Network Address Translation

Network Address Translation (NAT) enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a device, usually connecting two networks, and translates private (not globally unique) addresses in the internal network into legal addresses before packets are forwarded to another network. NAT can be configured to advertise only one address for the entire network to the outside world. A device configured with NAT will have at least one interface to the inside network and one to the outside network.

In a typical environment, NAT is configured at the exit device between a stub domain and the backbone. When a packet leaves the domain, NAT translates the locally significant source address to a global unique address. When a packet enters the domain, NAT translates the globally unique destination address into a local address. If more than one exit point exists, each NAT must have the same translation table. If the software cannot allocate an address because it has run out of addresses, it drops the packet and sends an Internet Control Message Protocol (ICMP) host unreachable packet.

With reference to NAT, the term “inside” refers to those networks that are owned by an organization and that must be translated. Inside this domain, hosts will have addresses in one address space. When NAT is configured and when the hosts are outside, hosts will appear to have addresses in another address space. The inside address space is referred to as the local address space and the outside address space is referred to as the global address space.

Consider a scenario where NAT translates both source and destination IP addresses. A packet is sent to a device from inside NAT with the source address 192.168.1.1 and the destination address 10.1.1.1. NAT translates these addresses and sends the packet to the external network with the source address 209.165.200.225 and the destination address 209.165.200.224.

Similarly, when the response comes back from outside NAT, the source address will be 209.165.200.225 and the destination address will be 209.165.200.224. Therefore, inside NAT, the packets will have a source address of 10.1.1.1 and a destination address of 192.168.1.1.

In this scenario, if you want to create an Application Control Engine (ACE) to be used in a firewall policy, the pre-NAT IP addresses (also known as inside local and outside global addresses) 192.168.1.1 and 209.165.200.224 must be used.

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 packets that arrive out of order to be copied and reassembled in the correct order. The OoO packet processing reduces the need to retransmit dropped packets and reduces the bandwidth needed for the transmission of traffic on a network. To configure OoO support, use the **parameter-map type ooo global** command.

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

OoO processing is not supported in Simple Mail Transfer Protocol (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 peer-to-peer protocol.
- FastTrack traffic peer-to-peer protocol.
- Gnutella Version 2 traffic peer-to-peer protocol.
- H.323 VoIP Protocol Version 4.
- HTTP—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 is shared.
- ICQ IM Protocol.
- Kazaa Version 2 peer-to-peer protocol.
- Match Protocol SIP—Match Protocol SIP.
- MSN Messenger IM protocol.
- POP3—Protocol that client e-mail applications use to retrieve mail from a mail server.
- SUNRPC—Sun RPC.
- 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 IPS and zone-based policy firewall with Layer 4 inspection are enabled.

Intrazone Support in the Zone-Based Firewall Application

Intrazone support allows a zone configuration to include users both inside and outside a network. Intrazone support allows traffic inspection between users belonging to the same zone but different networks. Traffic within the same zone cannot be inspected prior to Cisco IOS Release 15.0(1)M. 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 Firewalls

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). Perform 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

Use the following 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, a protocol, or a 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 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**class-map type inspect** [**match-any** | **match-all**] *class-map-name* **Example:** `Device(config)# class-map type inspect match-all c1`	Creates a Layer 3 or Layer 4 inspect type class map and enters class-map configuration mode.
Step 4	**match access-group** {*access-group* | **name** *access-group-name*} **Example:** `Device(config-cmap)# match access-group 101`	Configures the match criterion for a class map based on the access control list (ACL) name or number.
Step 5	**match protocol** *protocol-name* [**signature**] **Example:** `Device(config-cmap)# match protocol http`	Configures the match criterion for a class map on the basis of a specified protocol. • Only Cisco stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps. • **signature—**Signature-based classification for peer-to-peer packets is enabled.
Step 6	**match class-map** *class-map-name* **Example:** `Device(config-cmap)# match class-map c1`	Specifies a previously defined class as the match criteria for a class map.
Step 7	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.
Step 8	**show policy-map type inspect zone-pair session** **Example:** `Device(config-cmap)# show policy-map type inspect zone-pair session`	(Optional) Displays Cisco stateful packet inspection sessions created because a policy map is applied on the specified zone pair. **Note** The information displayed under the class-map field is the traffic rate (bits per second) of the traffic that belongs to the connection-initiating traffic only. Unless the connection setup rate is significantly high and is 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 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. **drop** [**log**]
7. **pass**
8. **service-policy type inspect** *policy-map-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p1`	Creates a Layer 3 and Layer 4 inspect type policy map and enters policy-map configuration mode.
Step 4	**class type inspect** *class-name* **Example:** `Device(config-pmap)# class type inspect c1`	Specifies the traffic class on which an action to perform and enters policy-map class configuration mode.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect inspect-params`	Enables Cisco stateful packet inspection.

	Command or Action	Purpose
Step 6	**drop** [**log**] **Example:** `Device(config-pmap-c)# drop`	(Optional) Drops packets that are matched with the defined class. **Note** Actions **drop** and **pass** are exclusive, and actions **inspect** and **drop** are exclusive; that is, you cannot specify both of them at the same time.
Step 7	**pass** **Example:** `Device(config-pmap-c)# pass`	(Optional) Allows packets that are matched with the defined class.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-pmap-c)# service-policy type inspect p1`	Attaches a firewall policy map to a zone pair.
Step 9	**end** **Example:** `Device(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Configuring a Parameter Map

Depending on your policy, you can configure either an inspect, URL filter, or a protocol-specific parameter map. If you configure a URL filter type or a protocol-specific policy, you must configure a parameter map. 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 connections that are allowed in the firewall after you change the parameter map. To clear existing connections, use the **clear zone-pair inspect sessions** command.

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

Creating an Inspect 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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect** {*parameter-map-name* | **global** | **default**} **Example:** `Device(config)# parameter-map type inspect eng-network-profile`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters that pertains to the **inspect** action and enters parameter map type inspect configuration mode.
Step 4	**log** {**dropped-packets** {**disable** | **enable**} | **summary** [**flows** *number*] [**time-interval** *seconds*]}	(Optional) Configures packet logging during the firewall activity.

	Command or Action	Purpose
	Example: `Device(config-profile)# log summary flows 15 time-interval 30`	**Note** This command is visible in parameter map type inspect configuration mode only.
Step 5	**alert** {**on** | **off**} **Example:** `Device(config-profile)# alert on`	(Optional) Enables Cisco stateful packet inspection alert messages that are displayed on the console.
Step 6	**audit-trail** {**on** | **off**} **Example:** `Device(config-profile)# audit-trail on`	(Optional) Enables audit trail messages.
Step 7	**dns-timeout** *seconds* **Example:** `Device(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 be managed while there is no activity).
Step 8	**icmp idle-timeout** *seconds* **Example:** `Device(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:** `Device(config-profile)# max-incomplete low 800`	(Optional) Defines the number of existing half-open sessions that will cause the Cisco firewall to start and stop deleting half-open sessions.
Step 10	**one-minute** {**low** | **high**} *number-of-connections* **Example:** `Device(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:** `Device(config-profile)# sessions maximum 200`	(Optional) Sets the maximum number of allowed sessions that can exist on a zone pair. • Use this command to limit the bandwidth used by the sessions.
Step 12	**tcp finwait-time** *seconds* **Example:** `Device(config-profile)# tcp finwait-time 5`	(Optional) Specifies the length of time a TCP session will be managed after the Cisco firewall detects a finish (FIN)-exchange.
Step 13	**tcp idle-time** *seconds* **Example:** `Device(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:** `Device(config-profile)# tcp max-incomplete host 500 block-time 10`	(Optional) Specifies threshold and blocking time values for TCP host-specific Denial-of-Service (DoS) detection and prevention.
Step 15	**tcp synwait-time** *seconds* **Example:** `Device(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:** `Device(config-profile)# tcp window-scale-enforcement 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 policy firewall.
Step 17	**udp idle-time** *seconds* **Example:** `Device(config-profile)# udp idle-time 75`	(Optional) Configures an idle timeout of UDP sessions that are going through the firewall.
Step 18	**end** **Example:** `Device(config-profile)# end`	Exits parameter map type inspect configuration mode and returns to privileged EXEC configuration mode.

Creating a URL Filter Parameter Map

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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type urlfilter** *parameter-map-name* **Example:** `Device(config)# parameter-map type urlfilter eng-network-profile`	Creates or modifies a parameter map for URL filtering parameters and enters parameter map type inspect configuration mode. **Note** This command is hidden depending on your release, but it continues to work. The **parameter-map type urlfpolicy** command can also be used to create URL filtering parameters for local, trend, Websense Internet filtering, and the N2H2 Internet blocking program. Depending on your release, use the URL filter policy rather than the URL filter action. 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 for more information.

	Command or Action	Purpose
Step 4	**alert** {**on** | **off**} **Example:** `Device(config-profile)# alert on`	(Optional) Enables Cisco stateful packet inspection alert messages that are displayed on the console.
Step 5	**allow-mode** {**on** | **off**} **Example:** `Device(config-profile)# allow-mode on`	(Optional) Enables the default mode of the filtering algorithm.
Step 6	**audit-trail** {**on** | **off**} **Example:** `Device(config-profile)# audit-trail on`	(Optional) Enables audit trail messages.
Step 7	**cache** *number* **Example:** `Device(config-profile)# cache 5`	(Optional) Controls how the URL filter handles the cache it maintains for HTTP servers.
Step 8	**exclusive-domain** {**deny** | **permit**} *domain-name* **Example:** `Device(config-profile)# exclusive-domain permit cisco.com`	(Optional) Adds a domain name to or from the exclusive domain list so that the Cisco firewall does not have to send lookup requests to the vendor server.
Step 9	**max-request** *number-of-requests* **Example:** `Device(config-profile)# max-request 80`	(Optional) Specifies the maximum number of outstanding requests that exist at a time.
Step 10	**max-resp-pak** *number-of-requests* **Example:** `Device(config-profile)# max-resp-pak 200`	(Optional) Specifies the maximum number of HTTP responses that the Cisco 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:** `Device(config-profile)# server vendor n2h2 10.193.64.22 port 3128 outside retrans 9 timeout 8`	Specifies the URL filtering server.
Step 12	**source-interface** *interface-name* **Example:** `Device(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 (N2H2 or Websense).

	Command or Action	Purpose
Step 13	**end** **Example:** `Device(config-profile)# end`	Exits parameter map type inspect configuration mode and returns to privileged EXEC configuration mode.

Configuring a Layer 7 Protocol-Specific Parameter Map

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

Before You Begin

To enable name resolution, you must enable the **ip domain name** command and the **ip name-server** 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*}}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type protocol-info** *parameter-map-name* **Example:** `Device(config)# parameter-map type protocol-info ymsgr`	Defines an application-specific parameter map and enters parameter map type inspect configuration mode.

	Command or Action	Purpose
Step 4	**server** {**name** *string* [**snoop**] | **ip** {*ip-address* | **range** *ip-address-start ip-address-end*}} **Example:** `Device(config-profile)# server name example1.example.com`	Configures a set of domain name system (DNS) servers with which a given instant messenger application will interact. **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, issue the **server** command multiple times within the parameter map of an instant messenger. Multiple entries are treated cumulatively.
Step 5	**end** **Example:** `Device(config-profile)# end`	Exits parameter map type inspect configuration mode and returns to privileged EXEC configuration mode.

Troubleshooting Tips

To display details of an Instant Messenger (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

Note When you configure a TCP-based Layer 7 policy for Deep Packet Inspection (DPI), Out-of-Order (OoO) packet processing is enabled by default. Use the **parameter-map type ooo global** command to configure the OoO packet support parameters or to disable OoO processing. Depending on your release, OoO processing was enabled for zone-based firewall and for Intrusion Prevention System (IPS)-shared sessions with Layer 4 match (**match protocol tcp**, **match protocol http**), and for any TCP-based Layer 7 packet ordering.

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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type ooo global** **Example:** `Device(config)# parameter-map type ooo global`	Configures OoO processing and enters parameter map type inspect configuration mode.
Step 4	**tcp reassembly alarm** {**on** | **off**} **Example:** `Device(config-profile)# tcp reassembly alarm on`	Specifies the alert message configuration.
Step 5	**tcp reassembly memory limit** *memory-limit* **Example:** `Device(config-profile)# tcp reassembly memory limit 2048`	Specifies the OoO box-wide buffer size.
Step 6	**tcp reassembly queue length** *queue-length* **Example:** `Device(config-profile)# tcp reassembly queue length 45`	Specifies the OoO queue length per TCP flow.
Step 7	**tcp reassembly timeout** *time-limit* **Example:** `Device(config-profile)# tcp reassembly timeout 34`	Specifies the OoO queue reassembly timeout value.
Step 8	**end** **Example:** `Device(config-profile)# end`	Exits parameter map type inspect configuration mode and returns to privileged EXEC configuration mode.

Configuring Intrazone Support in the Zone-Based Firewall Applications

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone-pair security** *zone-pair-name* [**source** *source-zone-name* **destination** *destination-zone-name*]
4. **exit**
5. **policy-map type inspect** *policy-map-name*
6. **class-map type inspect** *protocol-name* {**match-any** | **match-all**} *class-map-name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone-pair security** *zone-pair-name* [**source** *source-zone-name* **destination** *destination-zone-name*] **Example:** `Device(config)# zone-pair security zonepair17 source zone8 destination zone8`	Specifies the name of the zone pair that is attached to an interface, the source zone for information passing, and the destination zone for information passing through this zone pair. • Enters security zone-pair configuration mode. **Note** To configure intrazone support, the source zone and the destination zone must be the same.
Step 4	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 5	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect my-pmap`	Specifies a policy map name and enters policy-map configuration mode.

	Command or Action	Purpose
Step 6	**class-map type inspect** *protocol-name* {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config-pmap)# class-map type inspect aol match-any cmap1`	Specifies the firewall class map protocol and name.
Step 7	**end** **Example:** `Device(config-pmap)# end`	Exits policy map configuration mode and returns to privileged EXEC configuration mode.

Configuring Layer 7 Protocol-Specific Firewall Policies

Configure Layer 7 policy maps if you need extra provisioning for Layer 7 inspection modules. It is not necessary that you configure all Layer 7 policy maps specified in this section.

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

Layer 7 Class Map and Policy Map Restrictions

- Deep packet inspection (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 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. The 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.
- Depending on your release, 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 device is reloaded.

Configuring an HTTP Firewall Policy

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

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*
18. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect http** [**match-any** | **match-all**] *class-map-name* **Example:** `Device(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 class-map configuration mode.
Step 4	**match response body java-applet** **Example:** `Device(config-cmap)# match response body java-applet`	(Optional) Identifies Java applets in an HTTP connection.
Step 5	**match req-resp protocol violation** **Example:** `Device(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:** `Device(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.
Step 7	**match req-resp header content-type** {**violation** | **mismatch** | **unknown**} **Example:** `Device(config-cmap)# match req-resp header content-type mismatch`	(Optional) Configures an HTTP class map based on the content type of the HTTP traffic.
Step 8	**match** {**request** | **response** | **req-resp**} **header** [*header-name*] **count gt** *number* **Example:** `Device(config-cmap)# match req-resp header count gt 16`	(Optional) Configures an HTTP firewall policy to permit or deny HTTP traffic on the basis of both request and response messages whose header count does not exceed the specified maximum number of fields.
Step 9	**match** {**request** | **response** | **req-resp**} **header** [*header-name*] **length gt** *bytes* **Example:** `Device(config-cmap)# match response header length gt 50000`	(Optional) Permits or denies HTTP traffic based on the length of the HTTP request header.

	Command or Action	Purpose
Step 10	**match request** {**uri** | **arg**} **length gt** *bytes* **Example:** `Device(config-cmap)# match request uri length gt 500`	(Optional) Configures an HTTP firewall policy to use the Uniform Resource Identifier (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:** `Device(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:** `Device(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:** `Device(config-cmap)# match req-resp header transfer-encoding compress`	(Optional) Permits or denies HTTP traffic according to the specified transfer encoding of the message.
Step 14	**match** {**request** | **response** | **req-resp**} **header** [*header-name*] **regex** *parameter-map-name* **Example:** `Device(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, **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 the "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

	Command or Action	Purpose
		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 and 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:** `Device(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:** `Device(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:** `Device(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.
Step 18	**end** **Example:** `Device(config-cmap)# end`	(Optional) Exits class map configuration mode and returns to privileged EXEC mode.

Configuring 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**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect http** *policy-map-name* **Example:** `Device(config)# policy-map type inspect http myhttp-policy`	Creates a Layer 7 HTTP policy map and enters policy-map configuration mode.
Step 4	**class-type inspect http** *http-class-name* **Example:** `Device(config-pmap)# class-type inspect http http-class`	Creates a class map for the HTTP protocol.
Step 5	**allow** **Example:** `Device(config-pmap)# allow`	(Optional) Allows a traffic class that matches the class.
Step 6	**log** **Example:** `Device(config-pmap)# log`	Generates log messages.

	Command or Action	Purpose
Step 7	**reset** **Example:** `Device(config-pmap)# reset`	(Optional) Resets a TCP connection if the data length of the Simple Mail Transfer Protocol (SMTP) body exceeds the value configured in the **class-map type inspect smtp** command.
Step 8	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring 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*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type urlfpolicy** {**local** | **n2h2** | **websense**} *parameter-map-name*	Configures the URL filter name related to the parameter map, which can include local, Websense, or N2H2

	Command or Action	Purpose
	Example: `Device(config)# parameter-map type urlfpolicy websense websense-param-map`	parameters and enters parameter map type inspect configuration mode.
Step 4	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter map type inspect configuration mode and returns to global 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:** `Device(config)# class-map type urlfilter websense websense-param-map`	Configures the class map for the URL filter and enters class-map configuration mode.
Step 6	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
Step 7	**policy-map type inspect urlfilter** *policy-map-name* **Example:** `Device(config)# policy-map type inspect urlfilter websense-policy`	Configures the URL filter policy and enters policy-map configuration mode.
Step 8	**service-policy urlfilter** *policy-map-name* **Example:** `Device(config-pmap)# service-policy urlfilter websense-policy`	Applies the URL filter policy under the inspect class as the service policy.
Step 9	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring an IMAP Firewall Policy

Configuring an IMAP Class Map

Perform the following task to configure an Integrated Messaging Access Protocol (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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**reset**] [**secure-login**] [**timeout** *seconds*] **Example:** `Device(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:** `Device(config)# class-map type inspect imap imap-class`	Creates a class map for IMAP to enter the match criterion and enters class-map configuration mode.
Step 5	**log** **Example:** `Device(config-cmap)# log`	Generates log messages.
Step 6	**match invalid-command** **Example:** `Device(config-cmap)# match invalid-command`	(Optional) Locates invalid commands on an IMAP connection.

	Command or Action	Purpose
Step 7	**match login clear-text** **Example:** `Device(config-cmap)# match login clear-text`	(Optional) Locates nonsecure login when an IMAP server is used.
Step 8	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC configuration mode.

Configuring 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**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect imap** *policy-map-name* **Example:** `Device(config)# policy-map type inspect imap myimap-policy`	Creates a Layer 3 Integrated Messaging Access Protocol (IMAP) policy map and enters policy-map configuration mode.

	Command or Action	Purpose
Step 4	**class-type inspect imap** *imap-class-name* **Example:** `Device(config-pmap)# class-type inspect imap pimap`	Creates a class map for the IMAP protocol.
Step 5	**log** **Example:** `Device(config-pmap)# log`	Generates log messages.
Step 6	**reset** **Example:** `Device(config-pmap)# reset`	(Optional) Resets a TCP connection if the data length of the Simple Mail Transfer Protocol (SMTP) body exceeds the value that you configured in the **class-map type inspect smtp** command.
Step 7	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring an Instant Messenger Policy

Configuring an IM Class Map

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class map type inspect** {**aol** | **msnmsgr** | **ymsgr** | **icg** | **winmsgr**} [**match-any**] *class-map-name* **Example:** `Device(config)# class map type inspect aol myaolclassmap`	Creates an Instant Messenger (IM) type class map so that you can begin adding match criteria and enters class-map configuration mode.
Step 4	**match service** {**any** | **text-chat**} **Example:** `Device(config-cmap)# match service text-chat`	(Optional) Creates a match criterion on the basis of text chat messages.
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.

Configuring an IM Policy Map

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**
8. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy map type inspect** *protocol-name* *policy-map-name* **Example:** `Device(config)# policy map type inspect aol myaolpolicymap`	Creates an Instant Messenger (IM) policy map and enters policy-map configuration mode.
Step 4	**class type inspect** {**aol** | **msnmsgr** | **ymsgr** | **icq** | **winmsgr**} *class-map-name* **Example:** `Device(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 by using the **class-map type inspect** command.
Step 5	**reset** **Example:** `Device(config-pmap)# reset`	(Optional) Resets the connection.
Step 6	**log** **Example:** `Device(config-pmap)# log`	(Optional) Generates a log message for the matched parameters.
Step 7	**allow** **Example:** `Device(config-pmap)# allow`	(Optional) Allows the connection.
Step 8	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring a Peer-to-Peer Policy

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

Configuring a Peer-to-Peer Class Map

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*]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class map type inspect** {**edonkey** | **fasttrack** | **gnutella** | **kazaa2**} [**match-any**] *class-map-name* **Example:** `Device(config)# class map type inspect edonkey myclassmap`	Creates a peer-to-peer type class map so that you can begin adding match criteria and enters class-map configuration mode.
Step 4	**match file-transfer** [*regular-expression*] **Example:** `Device(config-cmap)# match file-transfer *`	(Optional) Matches file transfer connections within any supported peer-to-peer protocol. **Note** To specify that all file transfer connections should be identified by the traffic class, use "*" as the regular expression.
Step 5	**match search-file-name** [*regular-expression*] **Example:** `Device(config-cmap)# match search-file-name`	(Optional) Blocks filenames within a search request for clients using the eDonkey application. **Note** This command is applicable only for the eDonkey application.
Step 6	**match text-chat** [*regular-expression*] **Example:** `Device(config-cmap)# match text-chat`	(Optional) Blocks text chat messages between clients using the eDonkey peer-to-peer application. **Note** This command is applicable only for the eDonkey application.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.

Configuring a Peer-to-Peer Policy Map

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**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy map type inspect p2p** *policy-map-name* **Example:** `Device(config)# policy map type inspect p2p mypolicymap`	Creates a peer-to-peer policy map and enters policy-map configuration mode.
Step 4	**class type inspect** {**edonkey** | **fasttrack** | **gnutella** | **kazaa2**} *class-map-name* **Example:** `Device(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 in the **class-map type inspect** command.

	Command or Action	Purpose
Step 5	**reset** **Example:** `Device(config-pmap)# reset`	(Optional) Resets the connection.
Step 6	**log** **Example:** `Device(config-pmap)# log`	(Optional) Generates a log message for the matched parameters.
Step 7	**allow** **Example:** `Device(config-pmap)# allow`	(Optional) Allows the connection.
Step 8	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring a POP3 Firewall Policy

Configuring 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**
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**reset**] [**secure-login**] [**timeout** *seconds*] **Example:** `Device(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:** `Device(config)# class-map type inspect pop3 pop3-class`	Creates a class map for the Post Office Protocol, Version 3 (POP3) protocol to enter match criteria and enters class-map configuration mode.
Step 5	**match invalid-command** **Example:** `Device(config-cmap)# match invalid-command`	(Optional) Locates invalid commands on a POP3 server.
Step 6	**match login clear-text** **Example:** `Device(config-cmap)# match login clear-text`	(Optional) Locates a nonsecure login when using a POP3 server.
Step 7	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.

Configuring 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**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect pop3** *policy-map-name* **Example:** `Device(config)# policy-map type inspect pop3 mypop3-policy`	Creates a Layer 7 Post Office Protocol, Version 3 (POP3) policy map and enters policy-map configuration mode.
Step 4	**class-type inspect pop3** *pop3-class-name* **Example:** `Device(config-pmap)# class-type inspect pop3 pcl`	Creates a class map for the POP3 protocol.
Step 5	**log** **Example:** `Device(config-pmap)# log`	Generates log messages.
Step 6	**reset** **Example:** `Device(config-pmap)# reset`	(Optional) Resets a TCP connection if the data length of the Simple Mail Transfer Protocol (SMTP) body exceeds the value that you configured in the **class-map type inspect smtp** command.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring an SMTP Firewall Policy

Configuring 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 for more information on using this command.

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*
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**class-map type inspect smtp** [**match-all** | **match-any**] *class-map-name* **Example:** `Device(config)# class-map type inspect smtp smtp-class`	Creates a class map for the Simple Mail Transfer Protocol (SMTP) protocol to enter match criteria and enters class-map configuration mode.
Step 4	**match data-length gt** *max-data-value* **Example:** `Device(config-cmap)# match data-length gt 200000`	Determines if the amount of data transferred in an SMTP connection is above the configured limit.
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.

Configuring 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**
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**policy-map type inspect smtp** *policy-map-name* **Example:** `Device(config)# policy-map type inspect smtp mysymtp-policy`	Creates a Layer 7 Simple Mail Transfer Protocol (SMTP) policy map and enters policy-map configuration mode.
Step 4	**class-type inspect smtp** *smtp-class-name* **Example:** `Device(config-pmap)# class-type inspect smtp sc`	Configures inspection parameters for an SMTP protocol.
Step 5	**reset** **Example:** `Device(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.
Step 6	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring a SUNRPC Firewall Policy

Note If you are inspecting a remote-procedure call (RPC) protocol (that is, you have 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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect sunrpc** [**match-any**] *class-map-name* **Example:** `Device(config)# class-map type inspect sunrpc long-urls`	Creates a class map for the SUNRPC protocol to enter match criteria and enters class-map configuration mode.
Step 4	**match program-number** *program-number* **Example:** `Device(config-cmap)# match program-number 2345`	(Optional) Specifies the allowed remote-procedure call (RPC) protocol program number as a match criterion.
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring 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*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect sunrpc** *policy-map-name* **Example:** `Device(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:** `Device(config-pmap)# class-type inspect sunrpc cs1`	Configures inspection parameters for the SUNRPC protocol.
Step 5	**allow** [**wait-time** *minutes*] **Example:** `Device(config-pmap)# allow wait-time 10`	(Optional) Allows the configured program number. • Specifies the wait time in minutes to keep a keyhole open in the firewall to allow subsequent connections from the same source address to the same destination address and port. The default wait time is zero minutes. This keyword is available only for the remote-procedure call (RPC) protocol.
Step 6	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring an MSRPC Firewall Policy

Note If you are inspecting an remote-procedure call (RPC) protocol (that is, you have 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 the following 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**parameter-map type protocol-info msrpc** *parameter-map-name* **Example:** `Device(config)# parameter-map type protocol-info msrpc para-map`	Defines an application-specific parameter map and enters parameter map type inspect configuration mode.
Step 4	**timeout** *seconds* **Example:** `Device(config-profile)# timeout 60`	Configures the MSRPC endpoint mapper (EPM) timeout.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter map type inspect configuration mode and returns to global configuration mode.
Step 6	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any c-map`	Creates an inspect type class map for the traffic class and enters class-map configuration mode.
Step 7	**match protocol msrpc** **Example:** `Device(config-cmap)# match protocol msrpc`	Configures match criteria for a class map on the basis of a specified protocol. • Only Cisco stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 8	**match protocol msrpc-smb-netbios** **Example:** `Device(config-cmap)# match protocol msrpc-smb-netbios`	Configures match criteria for a class map on the basis of a specified protocol. • Only Cisco stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 9	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
Step 10	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p-map`	Creates a Layer 3 and Layer 4 inspect type policy map and enters policy-map configuration mode.
Step 11	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect c-map`	Specifies the traffic (class) on which an action is to be performed and enters policy-map class configuration mode.

	Command or Action	Purpose
Step 12	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables Cisco stateful packet inspection.
Step 13	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 14	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies the matching of the system default class and enters policy-map class configuration mode. • If the system default class is not specified, unclassified packets are matched.
Step 15	**drop** **Example:** `Device(config-pmap-c)# drop`	Drops packets that match a defined class.
Step 16	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 17	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode and returns to global configuration mode.
Step 18	**zone security** *security-zone-name* **Example:** `Device(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:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 20	**zone security** *security-zone-name* **Example:** `Device(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:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 22	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security in-out source in-zone destination out-zone`	Creates a zone pair and enters security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 23	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# 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:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to 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:

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

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.

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*
14. **platform inspect match-statistics per-filter**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *zone-name* **Example:** `Device(config)# zone security z1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**description** *line-of-description* **Example:** `Device(config-sec-zone)# description Internet Traffic`	(Optional) Describes the zone.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 6	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 7	**description** *line-of-description* **Example:** `Device(config-sec-zone-pair)# description accounting network`	(Optional) Describes the zone pair.
Step 8	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Configures an interface and enters interface configuration mode.
Step 10	**zone-member security** *zone-name* **Example:** `Device(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 in and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you should apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**zone-pair security** *zone-pair-name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode.

	Command or Action	Purpose
Step 13	**service-policy type inspect** *policy-map-name* **Example:** `Device(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.
Step 14	**platform inspect match-statistics per-filter** **Example:** `Device(config-sec-zone-pair)# platform inspect match-statistics per-filter`	Enables zone-based firewall per-filter statistics. **Note** To enable per-filter statistics on the device, do the following: • RELOAD the device. • OR Remove all the service-policies and re-apply the changes to the statistics. To activate the **platform inspect match-statistics per-filter** command, re-apply all service-policies.
Step 15	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to privileged EXEC mode.

Configuration Examples for Zone-Based Policy Firewalls

Example: Configuring Layer 3 and Layer 4 Firewall Policies

The following example shows a Layer 3 or Layer 4 top-level policy. The traffic is matched to the access control list (ACL) 199 and deep-packet HTTP inspection is configured. Configuring the **match access-group 101** enables Layer 4 inspection. As a result, Layer 7 inspection is omitted unless the class-map is of type **match-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: Adding WAN to self-zone and self-zone to WAN

The following example shows that a policy is not required to pass all Layer 2 Tunneling Protocol (L2TP) traffic to a router as the traffic allowed is destined to the router or the traffic is originated from the router.

However, in case we do not want all traffic to pass on to the router, and a policy is required to be configured for self-zone, we add WAN to the self-zone and self-zone to WAN to allow the L2TP traffic.

To allow the L2TP traffic, we need to use the below ACL in the classmap for the L2TP traffic:

```
ip access-list extended wan-self-pass
 permit udp any host 192.168.255.254 eq 1701

ip access-list extended self-wan-pass
 permit udp host 192.168.255.254 eq 1701 any
```

Example: Configuring Layer 7 Protocol-Specific Firewall Policies

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

```
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 Extended SMTP (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
```

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

Example: Configuring a URL Filter Policy

```
parameter-map type urlfpolicy websense-param-map
class-map type urlfilter websense websense-param-map
policy-map type inspect urlfilter websense-policy
service-policy urlfilter websense-policy
```

Example: Configuring a URL Filter Policy for Websense

Example: Websense Server Configuration

```
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

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

Example: Configuring 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: Creating Security Zones and Zone Pairs and Attaching a Policy Map to a Zone Pair

Example: Creating a Security Zone

The following example shows how to create security zone z1, which is called finance department networks, and security zone z2, which is called engineering services network:

```
zone security z1
 description finance department networks
!
zone security z2
 description engineering services network
```

Example: Creating Zone Pairs

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

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

Example: Assigning an Interface to a Security Zone

The following example shows how to attach Ethernet interface 0 to zone z1 and Ethernet interface 1 to zone z2:

```
interface ethernet0
 zone-member security z1
!
interface ethernet1
 zone-member security z2
```

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 are displayed in the **show policy-map type inspect zone-pair** command output if the class map matches any class map:

```
Device# 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)=>(172.16.1.2:23) 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
```

Example: Zone-Based Firewall Per-filter Statistics

The following configuration example shows how to prevent memory shortage when a large number of firewall filters are created. To prevent memory shortage, you can enable the zone-based firewall per-filter statistics with the **platform inspect match-statistics per-filter** command. In the example, for each filter (ACL or UDP), there are statistics available for the number of packets and the number of bytes traversed through zone-based firewall.

```
Device# show policy-map type inspect zone-pair ogacl_zp
 Zone-pair: ogacl_zp
  Service-policy inspect : ogacl_pm
Class-map: ogacl_cm (match-any)
  Match: access-group name ogacl
        xxx packets, xxx bytes
      Match: protocol udp
        xxx packets, xxx bytes
```

Note Per-filter statistics are available only for match-any filters and are not applicable for match-all cases.

Note For Cisco IOS XE 16.3 and Cisco IOS XE 16.4 releases, to enable per-filter statistics, either reload the device or remove the service-policies and then reapply the service policies on the zone pair before the **platform inspect match-statistics per-filter** command is activated.

For Cisco IOS XE 3.17 release, you must save the configuration and reload the system to activate this command.

Note Similarly, to disable per-filter statistics, either reload the device or remove the service-policies and then reapply the service policies on the zone pair.

To check the TCAM memory used in a device, use the **show platform hardware qfp active classification feature-manager shm-stats-counter** command.

```
Device# show platform hardware qfp active classification feature-manager shm-stats-counter
    Shared Memory Information:
    Total shared memory size: 16777216
    Used shared memory size: 14703656
```

Note If traffic drops or per-filter statistics counters are not displayed, then probabilty is the TCAM shared memory used is more than 75% of the total TCAM.

Note If the shared memory used in the device is more than 75% of the capacity, the following warning message is displayed :

```
%CPP_FM-3-CPP_FM_TCAM_WARNING: SIP1: cpp_sp_svr: TCAM limit exceeded: Already used 75
percent shared memory for per-filter stats.
```

If the shared memory used in the device is 100%, the following warning message is displayed:

```
%CPP_FM-3-CPP_FM_TCAM_WARNING: SIP1: cpp_sp_svr: TCAM limit exceeded: Shared memory for
per-filter stats overflow!
```

Additional References for Zone-Based Policy Firewalls

Related Documents

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

Standards and RFCs

Standard & RFC	Title
RFC 1950	ZLIB Compressed Data Format Specification version 3.3
RFC 1951	DEFLATE Compressed Data Format Specification version 1.3
RFC 2616	Hypertext Transfer Protocol—HTTP/1.1

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 Firewalls

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

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

Table 1: Feature Information for Zone-Based Policy Firewalls

Feature Name	Releases	Feature Information
Application Inspection and Control for HTTP—Phase 2	12.4(9)T	The Application Inspection and Control for HTTP—Phase 2 feature extends support for HTTP application firewall policies. The following commands were introduced or modified by this feature: **regexmatch body regex**, **match header count**, **match header length**, **match header regex**, **match request length**, **match request**, **match response status-line regex**.

Feature Name	Releases	Feature Information
E-mail Inspection Engine	15.1(1)S	The E-mail Inspection Engine feature allows users to inspect POP3, IMAP, and E/SMTP e-mail traffic contained in SSL VPN tunneled connections that traverse the Cisco device.
P2P Application Inspection and Control—Phase 1	12.4(9)T 12.4(20)T 15.3(1)T	The P2P Application Inspection and Control—Phase 1 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 (IM) 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 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**. In 15.3(1)T and later releases, the following peer-to-peer protocols are deprecated: • BitTorrent • DirectConnect • eDonkey • FastTrack • Gnutella Version 2 • Kazaa Version 2 • WinMX

Feature Name	Releases	Feature Information
Rate-Limiting Inspected Traffic	12.4(9)T	The Rate-Limiting Inspected Traffic feature allows users to rate limit traffic within a Cisco firewall (inspect) policy. Also, users can limit the absolute number of sessions that can exist on a zone pair. The following commands were introduced by this feature: **police (zone policy)** and **sessions maximum**.
Zone-Based Policy Firewalls	12.4(6)T	The Zone-Based Policy Firewall feature provides a Cisco 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—Default Zone	15.6(1)T	The Zone-Based Firewall— Default Zone feature introduces a default zone that enables a firewall policy to be configured on a zone pair that consist of a zone and a default zone. Any interface without explicit zone membership belongs to a default zone. The following commands were introduced by this feature: **zone pair security**, **zone security default**.
Zone-Based Firewall Support for Microsoft Remote Procedure Call (MSRPC)	15.1(4)M	The Zone-Based Firewall Support for MSRPC feature introduces zone-based policy firewall support for MSRPC.
Zone-Based Firewall Support of Multipoint TCP	15.4(3)M	Multipoint TCP seamlessly works with zone-based firewall Layer 4 inspection. Multipoint TCP does not work with application layer gateways (ALGs) and application inspection and control (AIC).

Feature Name	Releases	Feature Information
Zone-Based Firewall Usability and Manageability	15.0(1)M 15.1(1)T	The Zone-Based Firewall Usability and Manageability features covered in this document are out-of-order (OoO) packet processing support in zone-based firewalls, intrazone support in zone-based firewalls, and enhanced debug capabilities. 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**. Depending on your release, 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**.

CHAPTER 2

Zone-Based Policy Firewall IPv6 Support

The zone-based policy firewall IPv6 support feature coexists with the zone-based policy firewall for IPv4 in order to support IPv6 traffic. The feature provides MIB support for TCP, UDP, ICMPv6, and FTP sessions. This document describes how to configure parameter-maps, and to create and use class maps, policy maps, zones and zone pairs.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Information About Zone-Based Policy Firewall IPv6 Support

Zone-Based Policy Firewall IPv6 Support

The zone-based policy firewall for IPv6 coexists with the zone-based policy firewall for IPv4 in order to support IPv6 traffic. The feature provides MIB support for TCP, UDP, ICMPv6, and FTP sessions.

How to Configure Zone-Based Policy Firewall IPv6 Support

Configuring an Inspect-Type Parameter Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** {*parameter-map-name* | **global** | **default**}
4. **sessions maximum** *sessions*
5. **ipv6 routing-enforcement-header loose**

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 v6-param-map`	Configures an inspect type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the inspect action, and places the router in parameter map configuration mode.
Step 4	**sessions maximum** *sessions* **Example:** `Router(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 5	**ipv6 routing-enforcement-header loose** **Example:** `Router(config-profile)# ipv6 routing-enforcement-header loose`	Provides backward compatibility with legacy IPv6 inspection.

Creating and Using an Inspect-Type Class Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
4. **match protocol tcp**
5. **match protocol udp**
6. **match protocol icmp**
7. **match protocol ftp**

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** {**match-any** | **match-all**} *class-map-name* **Example:** `Router(config-profile)# class-map type inspect match-any v6-class`	Create an inspect type class map, and places the router in lass-map configuration mode.
Step 4	**match protocol tcp** **Example:** `Router(config-cmap)# match protocol tcp`	Configures the match criterion for a class map based on TCP.
Step 5	**match protocol udp** **Example:** `Router(config-cmap)# match protocol udp`	Configures the match criterion for a class map based on UDP.

	Command or Action	Purpose
Step 6	**match protocol icmp** **Example:** `Router(config-cmap)# match protocol icmp`	Configures the match criterion for a class map based on ICMP.
Step 7	**match protocol ftp** **Example:** `Router(config-cmap)# match protocol ftp`	Configures the match criterion for a class map based on FTP.

Creating and Using an Inspect-Type Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class type inspect** *class-map-name*
5. **inspect** [*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	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect v6-policy`	Creates an inspect-type policy map, and places the router in policy-map configuration mode.

	Command or Action	Purpose
Step 4	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect v6-class`	Specifies the traffic (class) on which an action is to be performed.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Router(config-pmap)# inspect`	Enables Cisco IOS stateful packet inspection.

Creating Security Zones and Zone Pairs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **zone security** {*zone-name* | **default**}
5. **zone-pair security** *zone-pair-name* **source** {*source-zone-name* | **self** | **default**} **destination** {*destination-zone-name* | **self** | **default**}
6. **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* | **default**}	Creates a security zone.

	Command or Action	Purpose
	Example: `Router(config)# zone security 1`	• Cisco recommends that you create at least two security zones so that you can create a zone pair.
Step 4	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security 2`	Creates a security zone. • Cisco recommends that you create at least two security zones so that you can create a zone pair.
Step 5	**zone-pair security** *zone-pair-name* **source** {*source-zone-name* | **self** | **default**} **destination** {*destination-zone-name* | **self** | **default**} **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair, and places the router in zone-pair configuration mode.
Step 6	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect v6-policy`	Attaches a firewall policy map to a zone pair.

Configuration Examples for Zone-Based Policy Firewall IPv6 Support

Example: Configuring Cisco IOS Zone-Based Firewall for IPv6

```
parameter-map type inspect v6-param-map
 sessions maximum 10000
 ipv6 routing-header-enforcement loose
!
!
class-map type inspect match-any v6-class
 match protocol tcp
 match protocol udp
 match protocol icmp
 match protocol ftp
!
!
policy-map type inspect v6-policy
 class type inspect v6-class
  inspect
!
```

```
zone security z1
zone security z2
!
zone-pair security zp source z1 destination z2
 service-policy type inspect v6-policy
```

Additional References for Zone-Based Policy Firewall IPv6 Support

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

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

Feature Information for Zone-Based Policy Firewall IPv6 Support

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

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

Table 2: Feature Information for Zone-Based Policy Firewall IPv6 Support

Feature Name	Releases	Feature Information
Zone-Based Policy Firewall IPv6 Support	15.1(2)T	Cisco zone-based firewall for IPv6 coexists with Cisco zone-based firewall for IPv4 in order to support IPv6 traffic.

CHAPTER 3

VRF-Aware Cisco Firewall

VRF-Aware Cisco Firewall applies Cisco Firewall functionality to Virtual Routing and Forwarding (VRF) interfaces when the firewall is configured on a service provider (SP) or large enterprise edge device. SPs can provide managed services to small and medium business markets.

The VRF-Aware Cisco Firewall supports VRF-aware URL filtering and VRF-lite (also known as Multi-VRF CE).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for VRF-Aware Cisco Firewall

- Understand Cisco firewalls.

- Configure VRFs.
- Verify that VRFs are operational.

Restrictions for VRF-Aware Cisco Firewall

- VRF-Aware Cisco Firewall is not supported on Multiprotocol Label Switching (MPLS) interfaces.
- If two VPN networks have overlapping addresses, VRF-aware network address translation (NAT) is required for them to support VRF-aware firewalls.
- When crypto tunnels belong to multiple VPNs terminate on a single interface, you cannot apply per-VRF firewall policies.

Information About VRF-Aware Cisco Firewall

Cisco Firewall

Cisco firewall provides robust, integrated firewall and intrusion detection functionality for every perimeter of the network. Available for a wide range of Cisco software-based devices, Cisco firewall offers sophisticated security and policy enforcement for connections within an organization (intranet) and between partner networks (extranets), as well as for securing Internet connectivity for remote and branch offices.

Cisco firewall enhances existing Cisco security capabilities such as authentication, encryption, and failover, with state-of-the-art security features such as stateful, application-based filtering (context-based access control), defense against network attacks, per-user authentication and authorization, and real-time alerts.

Cisco firewall is configurable through Cisco ConfigMaker software, an easy-to-use Microsoft Windows 95, Windows 98, NT 4.0 based software tool.

Cisco firewall provides great value in addition to these benefits:

- Flexibility—Provides multiprotocol routing, perimeter security, intrusion detection, VPN functionality, and dynamic per-user authentication and authorization.
- Investment protection—Leverages existing multiprotocol device investment.
- Scalable deployment—Scales to meet bandwidth and performance requirements of any network.
- VPN support—Provides a complete VPN solution based on Cisco IPSec and other Cisco software-based technologies, including Layer 2 Tunneling Protocol (L2TP) tunneling and quality of service (QoS).

The VRF-aware Cisco firewall is different from the non-VRF-aware firewall because it does the following:

- Allows users to configure a per-VRF firewall. The firewall inspects IP packets that are sent and received within a VPN routing and forwarding (VRF).
- Allows service providers (SP) to deploy the firewall on the provider edge (PE) device.
- Supports overlapping IP address space, thereby allowing traffic from nonintersecting VRFs to have the same IP address.

- Supports per-VRF (not global) firewall command parameters and denial-of-service (DoS) parameters so that the VRF-aware firewall can run as multiple instances (with VRF instances) allocated to various VPN customers.
- Performs per-VRF URL filtering.
- Generates VRF-specific syslog messages that can be seen only by a particular VPN. These alerts and audit-trail messages allow network administrators to manage the firewall; that is, they can adjust firewall parameters, detect malicious sources and attacks, add security policies, and so forth. The VRF name is tagged to syslog messages being logged to the syslog server.

Both VRF-aware and non-VRF-aware firewalls now allow you to limit the number of firewall sessions. Otherwise, it would be difficult for VRFs to share device resources because one VRF may consume a maximum amount of resources, leaving few resources for other VRFs. That would cause the DoS to other VRFs. To limit the number of sessions, enter the **ipinspectname** command.

VRF

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

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

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

VRF-lite

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

Note VRF-lite interfaces must be Layer 3 interfaces.

VRF-lite includes these devices:

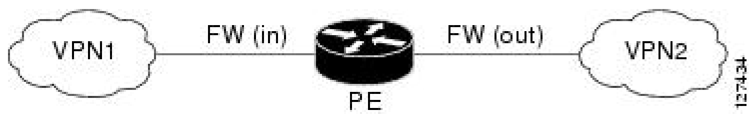
- Customer edge (CE) devices provide customer access to the service provider network over a data link to one or more provider edge (PE) routers. The CE device advertises the site's local routes to the PE router and learns the remote VPN routes from it. A Catalyst 4500 switch can be a CE.
- Provider edge (PE) routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.

- Provider routers (or core routers) are any routers in the service provider network that do not attach to CE devices.
- The PE is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE to maintain all of the service provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CEs, a PE router exchanges VPN routing information with other PE routers by using internal BGP (IBPG).

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

In a VRF-to-VRF situation, if firewall policies are applied on both inbound and outbound interfaces as shown in the figure below, the firewall on the inbound interface takes precedence over the firewall on the outbound interface. If the incoming packets do not match against the firewall rules (that is, the inspection protocols) configured on the inbound interface, the firewall rule on the outbound interface is applied to the packet.

Figure 4: Firewall in a VRF-to-VRF Scenario



Per-VRF URL Filtering

The VRF-aware firewall supports per-VRF URL filtering. Each VPN can have its own URL filter server. The URL filter server typically is placed in the shared service segment of the corresponding VPN. (Each VPN has a VLAN segment in the shared service network.) The URL filter server can also be placed at the customer site.

AlertsandAuditTrails

Context-based access control (CBAC) generates real-time alerts and audit trails based on events tracked by the firewall. Enhanced audit trail features use SYSLOG to track all network transactions; recording time stamps, the source host, the destination host, ports used, and the total number of transmitted bytes, for advanced, session-based reporting. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity. Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for HTTP traffic, you can specify that in the CBAC rule covering HTTP inspection.

MPLS VPN

The Multiprotocol Label Switching (MPLS) VPN Feature allows multiple sites to interconnect transparently through a service provider (SP)network. One SP network can support several IP VPNs. Each VPN appears to its users as a private network, separate from all other networks. Within a VPN, each site can send IP packets to any other site in the same VPN.

Each VPN is associated with one or more VPN routing and forwarding (VRF) instances. A VRF consists of an IP routing table, a derived Cisco Express Forwarding table, and a set of interfaces that use the forwarding table.

The device maintains a separate routing and Cisco Express Forwarding table for each VRF. This prevents information from being sent outside the VPN and allows the same subnet to be used in several VPNs without causing duplicate IP address problems.

The device using Multiprotocol BGP (MP-BGP) distributes the VPN routing information using the MP-BGP extended communities.

VRF-aware NAT

Network Address Translation (NAT) allows a single device, such as a router, to act as an agent between the Internet (or public network) and a local (or private) network. Although NAT systems can provide broad levels of security advantages, their main objective is to economize on address space.

NAT allows organizations to resolve the problem of IP address depletion when they have existing networks and need to access the Internet. Sites that do not yet possess NIC-registered IP addresses must acquire them. Cisco IOS NAT eliminates concern and bureaucratic delay by dynamically mapping thousands of hidden internal addresses to a range of easy-to-get addresses.

In general, a NAT system makes it more difficult for an attacker to determine the following:

- Number of systems running on a network
- Type of machines and operating systems they are running
- Network topology and arrangement

NAT integration with MPLS VPNs allows multiple MPLS VPNs to be configured on a single device to work together. NAT can differentiate which MPLS VPN it receives IP traffic from even if the MPLS VPNS are all using the same IP addressing scheme. This enables multiple MPLS VPN customers to share services while ensuring that each MPLS VPN is completely separate from the other.

MPLS service providers would like to provide value-added services such as Internet connectivity, domain name servers (DNS), and VoIP service to their customers. This requires that their customers IP addresses be different when reaching the services. Because MPLS VPN allows customers to use overlapped IP addresses in their networks, NAT must be implemented to make the services possible.

There are two approaches to implementing NAT in the MPLS VPN network. NAT can be implemented on the CE router, which is already supported by NAT, or it can be implemented on a PE router. The NAT Integration with MPLS VPNs feature enables the implementation of NAT on a PE router in an MPLS cloud.

VRF-aware IPSec

The VRF-aware IPSec feature maps an IP Security (IPSec) tunnel to an MPLS VPN. Using the VRF-aware IPSec feature, you can map IPSec tunnels to VRF instances using a single public-facing address.

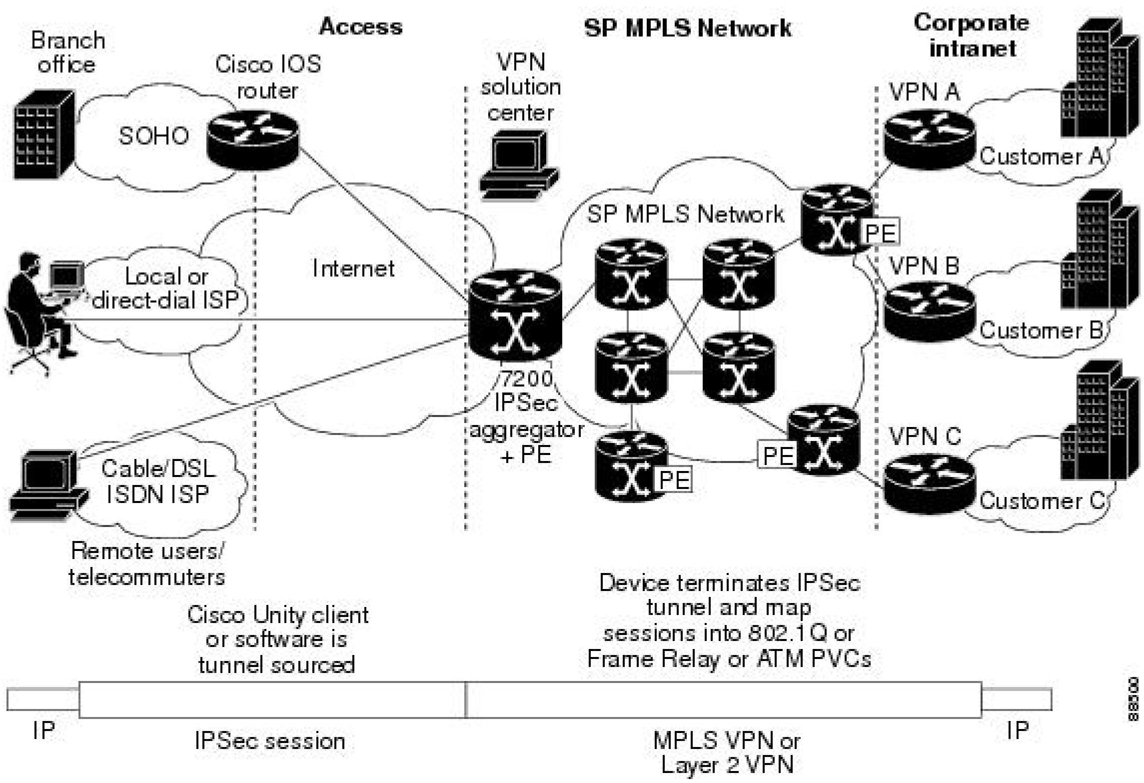
Each IPSec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to a VRF domain called the Front Door VRF (FVRF). The inner, protected IP packet belongs to a domain called the

Inside VRF (IVRF). In other words, the local endpoint of the IPSec tunnel belongs to the FVRF, whereas the source and destination addresses of the inside packet belong to the IVRF.

One or more IPSec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the Internet Security Association and Key Management Protocol (ISAKMP) profile that is attached to a crypto map entry.

The figure below illustrates a scenario showing IPSec to MPLS and Layer 2 VPNs.

Figure 5: IPSec-to-MPLS and Layer 2 VPNs



VRF Aware Cisco IOS Firewall Deployment

A firewall can be deployed at many points within the network to protect VPN sites from Shared Service (or the Internet) and vice versa. The following firewall deployments are described:

Distributed Network Inclusion of VRF Aware Cisco IOS Firewall

A VRF Aware Cisco IOS Firewall in a distributed network has the following advantages:

- The firewall is distributed across the MPLS core, so the firewall processing load is distributed to all ingress PE routers.
- VPN Firewall features can be deployed in the inbound direction.

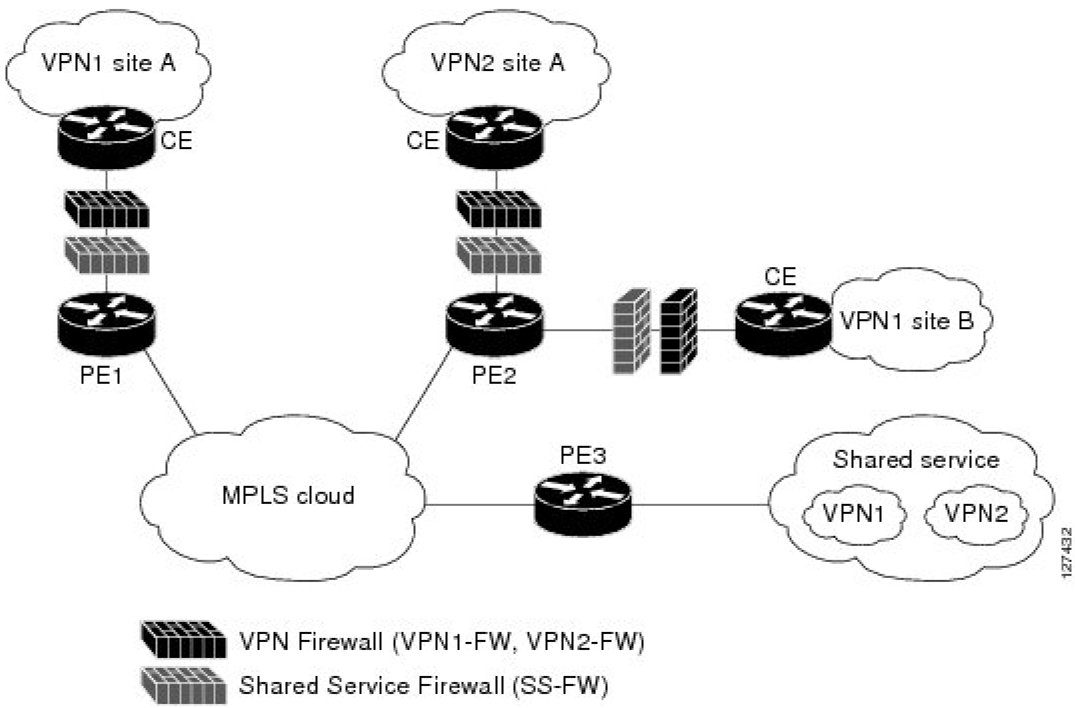
- Shared Service is protected from the VPN site at the ingress PE router; therefore, malicious packets from VPN sites are filtered at the ingress PE router before they enter the MPLS core.

However, the following disadvantages exist:

- There is no centralized firewall deployment, which complicates the deployment and management of the firewall.
- Shared Service firewall features cannot be deployed in the inbound direction.
- The MPLS core is open to the Shared Service. Therefore, malicious packets from Shared Service are filtered only at the ingress PE router after traveling through all core routers.

The figure below illustrates a typical situation in which an SP offers firewall services to VPN customers VPN1 and VPN2, thereby protecting VPN sites from the external network (for example, Shared Services and the Internet) and vice versa.

Figure 6: Distributed Network



In this example, VPN1 has two sites, Site A and Site B, that span across the MPLS core. Site A is connected to PE1, and Site B is connected to PE2. VPN2 has only one site that is connected to PE2.

Each VPN (VPN1 and VPN2) has the following:

- A VLAN segment in the Shared Service that is connected to the corresponding VLAN subinterface on PE3.
- Internet access through the PE3 router that is connected to the Internet

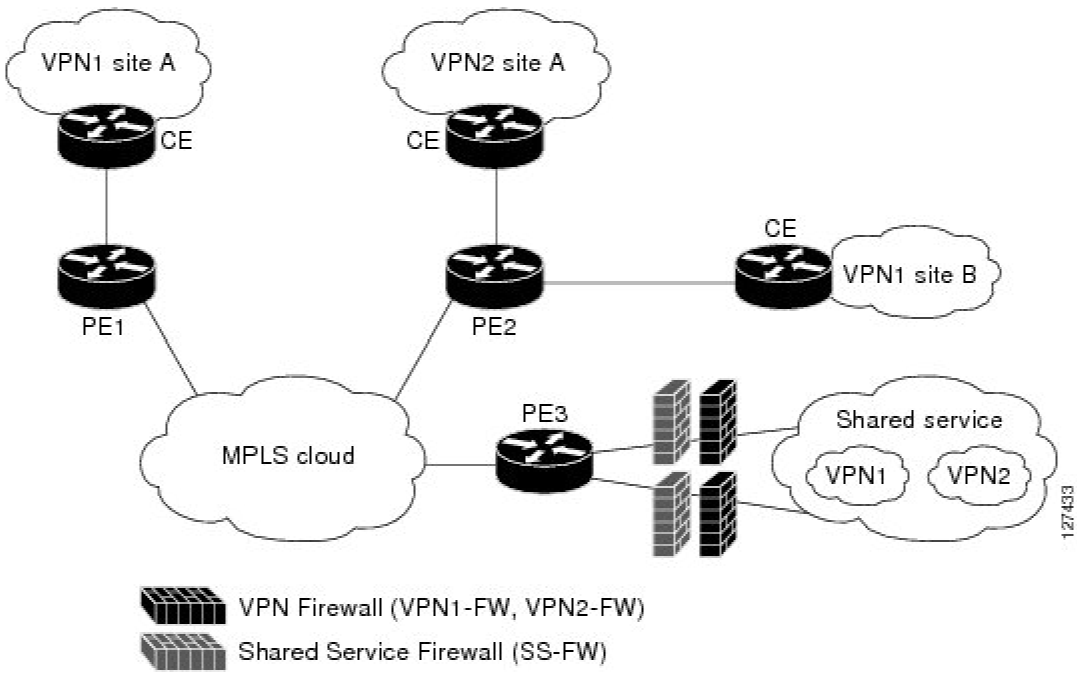
A distributed network requires the following firewall policies:

- VPN Firewall (VPN1-FW and VPN2-FW)--Inspects VPN-generated traffic that is destined to Shared Service or the Internet and blocks all non-firewall traffic that is coming from outside (Shared Service or the Internet), thereby protecting the VPN sites from outside traffic. This firewall typically is deployed on the VRF interface of the ingress PE router that is connected to the VPN site being protected. It is deployed in the inbound direction because the VRF interface is inbound to the VPN site being protected.
- Shared Service Firewall (SS-FW)--Inspects Shared Service-originated traffic that is destined to VPN sites and blocks all non-firewall traffic that is coming from outside (the VPN site), thereby protecting the Shared Service network from VPN sites. This firewall typically is deployed on the VRF interface of the ingress PE router that is connected to the VPN site from where the Shared Service is being protected. It is deployed in the outbound direction because the VRF interface is outbound to the Shared Service that is being protected.
- Generic-VPN Firewall (GEN-VPN-FW)--Inspects VPN-generated traffic that is destined to the Internet and blocks all non-firewall traffic that is coming from the Internet, thereby protecting all VPNs from the Internet. This firewall typically is deployed on the Internet-facing interface of the PE router that is connected to the Internet. It is deployed in the outbound direction because the Internet-facing interface is outbound to VPNs being protected.
- Internet Firewall (INET-FW)--Inspects Internet-generated traffic that is destined to Shared Service and blocks all non-firewall traffic that is coming from VPNs or Shared Service, thereby protecting the Internet from VPNs. This firewall typically is deployed on the Internet-facing interface of the PE router that is connected to the Internet. It is deployed in the inbound direction because the Internet-facing interface is inbound to the Internet being protected.

Hub-and-Spoke Network Inclusion of VRF Aware Cisco IOS Firewall

The figure below illustrates a hub-and-spoke network where the firewalls for all VPN sites are applied on the egress PE router PE3 that is connected to the Shared Service.

Figure 7: Hub-and-Spoke Network



Typically each VPN has a VLAN and/or VRF subinterface connected to the Shared Service. When a packet arrives from an MPLS interface, the inner tag represents the VPN-ID. MPLS routes the packet to the corresponding subinterface that is connected to Shared Service.

A Hub-and-Spoke network requires the following firewall policies:

- VPN Firewall (VPN1-FW and VPN2-FW)--Inspects VPN-generated traffic that is destined to Shared Service and blocks all non-firewall traffic that is coming from Shared Service, thereby protecting the VPN sites from Shared Service traffic. This firewall typically is deployed on the VLAN subinterface of the egress PE router that is connected to the Shared Service network. It is deployed in the outbound direction because the VLAN interface is outbound to the VPN site being protected.
- Shared Service Firewall (SS-FW)--Inspects Shared Service originated traffic that is destined to the VPN/Internet and blocks all non-firewall traffics that is coming from outside, thereby protecting the Shared Service network from VPN/Internet traffic. This firewall typically is deployed on the VLAN interface of the egress PE router that is connected to the Shared Service being protected. It is deployed in the inbound direction because the VLAN interface is inbound to the Shared Service being protected.
- Generic-VPN firewall (GEN-VPN-FW)--Inspects VPN-generated traffic that is destined to the Internet and blocks all non-firewall traffic that is coming from the Internet, thereby protecting all VPNs from the Internet. This firewall typically is deployed on the Internet-facing interface of the PE router that is connected to the Internet. It is deployed in the outbound direction because the Internet-facing interface is outbound to the VPNs being protected.

- Internet firewall (INET-FW)--Inspects Internet-generated traffic that is destined to Shared Service and blocks all non-firewall traffic that is coming from VPNs or Shared Service, thereby protecting the Internet from VPNs. This firewall typically is deployed on the Internet-facing interface of the PE router that is connected to the Internet. It is deployed in the inbound direction because the Internet-facing interface is inbound to the Internet being protected.

How to Configure VRF-Aware Cisco Firewall

Configuring and Checking ACLs to Ensure that Non-Firewall Traffic is Blocked

To configure access control lists (ACLs) and verify that only inspected traffic can pass through the firewall, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. **interface** *interface-type*
5. **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended vpn-acl`	Defines an extended IP ACL to block non-firewall traffic in both inbound and outbound directions.
Step 4	**interface** *interface-type* **Example:** `Device(config)# interface ethernet 0/1.10`	Enters interface configuration mode and specifies an interface that is associated with a VPN routing and forwarding (VRF).

	Command or Action	Purpose
Step 5	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**} **Example:** `Device(config-if)# ip access-group vpn-acl in`	Controls access to an interface. Applies the previously defined IP access list to a VRF interface whose non-firewall traffic is blocked.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Creating and Naming Firewall Rules and Applying the Rules to an Interface

To create and name firewall rules and apply the rules to an interface, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* [**parametermax-sessions***number*] *protocol* [**alert** {**on** \| **off**}] [**audit-trail** {**on** \| **off**}] [**timeout***seconds*]
4. **interface** *interface-id*
5. **ip inspect** *rule-name* {**in** \| **out**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* [**parametermax-sessions***number*] *protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout***seconds*]	Defines a set of inspection rules.

	Command or Action	Purpose
	Example: `Device(config)# ip inspect name vpn-fw ftp`	
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface ethernet 0/1.10`	Enters interface configuration mode and specifies an interface that is associated with a VPN routing and forwarding (VRF).
Step 5	**ip inspect** *rule-name* {**in** | **out**} **Example:** `Device(config-if)# ip inspect vpn-fw in`	Applies the previously defined inspection rule to a VRF interface whose traffic needs to be inspected.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Identifying and Setting Firewall Attributes

To identify and set firewall attributes, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect tcp max-incomplete host** *number* **block-time** *minutes* [**vrf***vrf-name*]
4. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip inspect tcp max-incomplete host** *number* **block-time** *minutes* [**vrf***vrf-name*] **Example:** `Device(config)# ip inspect tcp max-incomplete host 256 vrf bank-vrf`	Specifies threshold and blocking time values for TCP host-specific denial-of-service (DoS) detection and prevention.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Verifying the VRF-Aware Cisco Firewall Configuration and Functioning

Verify the configuration and functioning of the firewall by entering commands shown below.

SUMMARY STEPS

1. **show ip inspect** {**name***inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**}[**vrf***vrf-name*]
2. **show ip urlfilter** {**config** | **cache** | **statistics**} [**vrf***vrf-name*]

DETAILED STEPS

Step 1 **show ip inspect** {**name***inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**}[**vrf***vrf-name*]

Use this command to view firewall configurations, sessions, statistics, and so forth, pertaining to a specified VPN routing and forwarding (VRF). For example, to view firewall sessions pertaining to the VRF bank, enter the following command:

Example:

```
Device# show ip inspect interfaces vrf bank
```

Step 2 **show ip urlfilter** {**config** | **cache** | **statistics**} [**vrf***vrf-name*]

Use this command to view configurations, cache entries, statistics, and so forth, pertaining to a specified VRF. For example, to view the URL filtering statistics pertaining to the VRF bank, enter the following command:

Example:

```
Device# show ip urlfilter statistics vrf bank
```

Configuration Examples for VRF-Aware Cisco Firewall

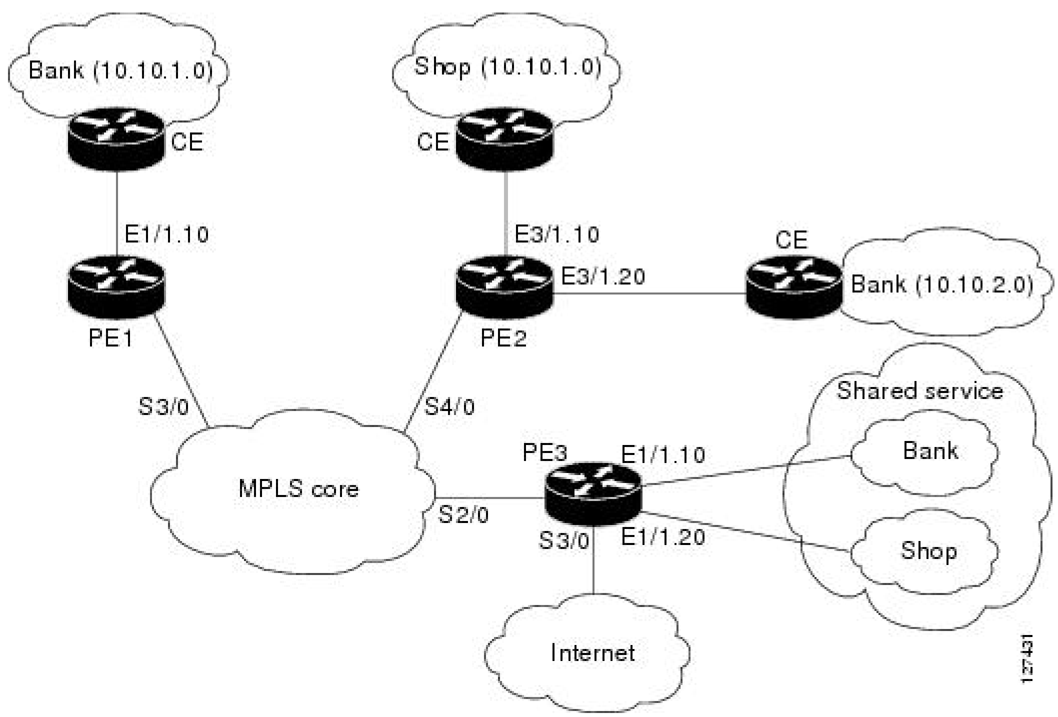
In the example illustrated in the figure below, a service provider (SP) offers firewall service to VPN customers Bank and Shop. The Bank VPN has the following two sites in a Multiprotocol Label Switching (MPLS) network:

- Site connected to PE1, whose network address is 10.10.1.0/24
- Site connected to PE2, whose network address is 10.10.2.0/24

The Bank VPN also has a VLAN network segment in shared service that is connected to PE3.

The Shop VPN has only one site, which is connected to PE4. The network address 10.10.1.0/24 is the same network address to which the Bank VPN site is connected.

Figure 8: VPN with Two Sites Across MPLS Network



Each VPN needs the following two firewalls:

- VPN firewall to protect the VPN site from shared services.
- Shared service firewall to protect shared service from the VPN site.

In addition, the following two firewalls are required:

- Internet firewall to protect VPNs from the Internet.
- Generic VPN firewall to protect the Internet from VPNs.

In this example, the security policies for Bank and Shop VPNs are as follows:

- Bank VPN firewall--bank_vpn_fw (Inspects FTP, HTTP, and ESMTP protocols)
- Bank shared service firewall--bank_ss_fw (Inspects ESMTP protocol)
- Shop VPN firewall--shop_vpn_fw (Inspects HTTP and RTSP protocols)
- Shop shared service firewall--shop_ss_fw (Inspects H323 protocol)

Security policies for the Internet firewall and generic VPN firewall are as follows:

- Internet firewall--inet_fw (Inspects HTTP and ESMTP protocols)
- Generic VPN firewall--gen_vpn_fw (Inspects FTP, HTTP, ESMTP, and RTSP protocols)

DISTRIBUTED NETWORK

PE1:

```
! VRF instance for the Bank VPN
ip vrf bank
rd 100:10
route-target export 100:10
route-target import 100:10

!
! VPN Firewall for Bank VPN protects Bank VPN from Shared Service
ip inspect name bank-vpn-fw ftp
ip inspect name bank-vpn-fw http
ip inspect name bank-vpn-fw esmtp

!
! Shared Service firewall for Bank VPN protects Shared Service from Bank VPN
ip inspect name bank-ss-fw esmtp

!
! VRF interface for the Bank VPN
interface ethernet 0/1.10

!
! description of VPN site Bank to PE1
encapsulation dot1Q 10
ip vrf forwarding bank
ip address 10.10.1.2 255.255.255.0
ip access-group bank-ss-acl in
ip access-group bank-vpn-acl out
ip inspect bank-vpn-fw in
ip inspect bank-ss-fw out

!
! MPLS interface
interface Serial3/0
 ip unnumbered Loopback0

 tag-switching ip

 serial restart-delay 0

!
! ACL that protects the VPN site Bank from Shared Service
ip access-list extended bank-vpn-acl

 permit ip 10.10.2.0 0.0.0.255 10.10.1.0 0.0.0.255

 permit tcp any any eq smtp

 deny   ip  any any log
```

```
!
! ACL that protects Shared Service from VPN site Bank
ip access-list extended bank-ss-acl

 permit ip 10.10.1.0 0.0.0.255 10.10.2.0 0.0.0.255

 permit tcp any any eq ftp

 permit tcp any any eq http
 permit tcp any any eq smtp

 deny   ip  any any log
```

PE2:

```
! VRF instance for the Bank VPN
ip vrf bank
rd 100:10
route-target export 100:10
route-target import 100:10

!
! VRF instance for the Shop VPN
ip vrf shop
rd 200:20
route-target export 200:20
route-target import 200:20

!
! VPN firewall for Bank BPN protects Bank VPN from Shared Service
ip inspect name bank-vpn-fw ftp
ip inspect name bank-vpn-fw http
ip inspect name bank-vpn-fw esmtp

!
! Shared Service firewall for Bank VPN protects Shared Service from Bank VPN
ip inspect name bank-ss-fw esmtp

!
! VPN firewall for Shop VPN protects Shop VPN from Shared Service
ip inspect name shop-vpn-fw http
ip inspect name shop-vpn-fw rtsp

!
! Shared Service firewall for Shop VPN protects Shared Service from Shop VPN
ip inspect name shop-ss-fw h323
!
! VRF interface for the Bank VPN
interface Ethernet 3/1.10

!
! description of VPN site Bank to PE2
encapsulation dot1Q 10
ip vrf forwarding bank
ip address 10.10.2.2 255.255.255.0
ip access-group bank-ss-acl in
ip access-group bank-vpn-acl out
ip inspect bank-vpn-fw in
ip inspect bank-ss-fw out

!
interface Ethernet 3/1.20

!
! description of VPN site Shop to PE2
encapsulation dot1Q 20
ip vrf forwarding shop
ip address 10.10.1.2 255.255.255.0
ip access-group shop-ss-acl in
```

```
ip access-group shop-vpn-acl out
ip inspect shop-vpn-fw in
ip inspect shop-ss-fw out
interface Serial 4/0

 ip unnumbered Loopback0

 tag-switching ip

 serial restart-delay 0

!
! ACL that protects the VPN site Bank from Shared Service
ip access-list extended bank-vpn-acl

 permit ip 10.10.1.0 0.0.0.255 10.10.2.0 0.0.0.255

 permit tcp any any eq smtp

 deny   ip  any any log

!
! ACL that protects Shared Service from VPN site Bank
ip access-list extended bank-ss-acl
 permit ip 10.10.2.0 0.0.0.255 10.10.1.0 0.0.0.255

 permit tcp any any eq ftp

 permit tcp any any eq http

 permit tcp any any eq smtp

 deny   ip  any any log

!
! ACL that protects VPN site Shop from Shared Service
ip access-list extended shop-vpn-acl

 permit tcp any any eq h323

 deny   ip any any log

!
ip access-list extended shop-ss-acl

 permit tcp any any eq http

 permit tcp any any eq rtsp
deny   ip  any any log
```

PE3:

```
! VRF instance for the Bank VPN
ip vrf bank
rd 100:10
route-target export 100:10
route-target import 100:10

!
! VRF instance for the Shop VPN
ip vrf shop
rd 200:20
route-target export 200:20
route-target import 200:20

!
! Generic VPN firewall to protect Shop and Bank VPNs from internet
ip inspect name gen-vpn-fw esmtp
ip inspect name gen-vpn-fw ftp
```

```
ip inspect name gen-vpn-fw http
ip inspect name gen-vpn-fw rtsp

!
! Internet firewall to prevent malicious traffic from being passed
! to internet from Bank and Shop VPNs
ip inspect name inet-fw esmtp
ip inspect name inet-fw http

!
! VRF interface for the Bank VPN
interface Ethernet 1/1.10

!
! Description of Shared Service to PE3
encapsulation dot1Q 10
ip vrf forwarding bank
ip address 10.10.1.50 255.255.255.0

!
! VRF interface for the Shop VPN
interface Ethernet 1/1.20

!
! Description of Shared Service to PE3
encapsulation dot1Q 20
ip vrf forwarding shop
ip address 10.10.1.50 255.255.255.0
interface Serial 2/0

 ip unnumbered Loopback0

 tag-switching ip

 serial restart-delay 0

!
! VRF interface for the Bank VPN
interface Serial 3/0

!
! Description of Internet-facing interface
ip address 192.168.10.2 255.255.255.0
ip access-group inet-acl out
ip access-group gen-vpn-acl in
ip inspect gen-vpn-fw out
ip inspect inet-fw in

!
! ACL that protects the Bank and Shop VPNs from internet
ip access-list extended gen-vpn-acl

 permit tcp any any eq smtp

 permit tcp any any eq www

 deny   ip  any any log

!
! ACL that protects internet from Bank and Shop VPNs
ip access-list extended inet-acl

 permit tcp any any eq ftp

 permit tcp any any eq http

 permit tcp any any eq smtp

 permit tcp any any eq rtsp

 deny   ip  any any log
```

HUB-AND-SPOKE NETWORK

PE3:

```
! VRF instance for the VPN Bank
ip vrf bank
rd 100:10
route-target export 100:10
route-target import 100:10

!
! VRF instance for the VPN Shop
ip vrf shop
rd 200:20
route-target export 200:20
route-target import 200:20

!
! VPN firewall for Bank BPN protects Bank VPN from Shared Service
ip inspect name bank-vpn-fw ftp
ip inspect name bank-vpn-fw http
ip inspect name bank-vpn-fw esmtp

!
! Shared Service firewall for Bank VPN protects Shared Service from Bank VPN
ip inspect name bank-ss-fw esmtp

!
! VPN firewall for Shop VPN protects Shop VPN from Shared Service
ip inspect name shop-vpn-fw http
ip inspect name shop-vpn-fw rtsp

!
! Shared Service firewall for Shop VPN protects Shared Service from Shop VPN
ip inspect name shop-ss-fw h323

!
! Generic VPN firewall protects Shop and Bank VPNs from internet
ip inspect name gen-vpn-fw esmtp
ip inspect name gen-vpn-fw ftp
ip inspect name gen-vpn-fw http
ip inspect name gen-vpn-fw rtsp

!
! Internet firewall prevents malicious traffic from being passed
! to internet from Bank and Shop VPNs
ip inspect name inet-fw esmtp
ip inspect name inet-fw http

!
! VRF interface for the Bank VPN
interface Ethernet 1/1.10

!
! description of Shared Service to PE3
encapsulation dot1Q 10
ip vrf forwarding bank
ip address 10.10.1.50 255.255.255.0
ip access-group bank-ss-acl out
ip access-group bank-vpn-acl in
ip inspect bank-vpn-fw out
ip inspect bank-ss-fw in

!
! VRF interface for the Shop VPN
interface Ethernet 1/1.20
!
! description of Shared Service to PE3
encapsulation dot1Q 20
```

```
ip vrf forwarding shop
ip address 10.10.1.50 255.255.255.0
ip access-group shop-ss-acl out
ip access-group shop-vpn-acl in
ip inspect shop-vpn-fw out
ip inspect shop-ss-fw in
interface Serial 2/0

 ip unnumbered Loopback0

 tag-switching ip

 serial restart-delay 0
!
! VRF interface for the Bank VPN
interface Serial 3/0

!
! description of Internet-facing interface
ip address 192.168.10.2 255.255.255.0
ip access-group inet-acl out
ip access-group gen-vpn-acl in
ip inspect gen-vpn-fw out
ip inspect inet-fw in

!
! ACL that protects the VPN site Bank from Shared Service
ip access-list extended bank-vpn-acl

 permit tcp any any eq smtp

 deny   ip  any any log
!
! ACL that protects Shared Service from VPN site Bank
ip access-list extended bank-ss-acl

 permit tcp any any eq ftp

 permit tcp any any eq http

 permit tcp any any eq smtp

 deny   ip  any any log

!
! ACL that protects VPN site Shop from Shared Service
ip access-list extended shop-vpn-acl

 permit tcp any any eq h323

 deny   ip any any log

!
ip access-list extended shop-ss-acl

 permit tcp any any eq http
 permit tcp any any eq rtsp
 deny   ip  any any log
!
! ACL that protects the Bank and Shop VPNs from internet
ip access-list extended gen-vpn-acl

 permit tcp any any eq smtp

 permit tcp any any eq www
 deny   ip  any any log
!
! ACL that protects internet from Bank and Shop VPNs
ip access-list extended inet-acl

 permit tcp any any eq ftp
```

```
 permit tcp any any eq http

 permit tcp any any eq smtp

 permit tcp any any eq rtsp

 deny   ip  any any log
```

In the example illustrated in the figure below, the Cisco firewall is configured on PE1 on the VPN routing and forwarding (VRF) interface E3/1. The host on NET1 wants to reach the server on NET2.

Figure 9: Sample VRF-Aware Cisco Firewall Network

The configuration steps are followed by a sample configuration and log messages.

1 Configure VRF on provider edge (PE) devices.
2 Ensure that your network supports MPLS traffic engineering.
3 Confirm that the VRF interface can reach NET1 and NET2.
4 Configure the VRF-aware Cisco firewall.
    1 Configure and apply access control lists (ACLs).
    2 Create firewall rules and apply them to the VRF interface.
5 Check for VRF firewall sessions.

VRF Configuration on PE1

```
! configure VRF for host1
ip cef
ip vrf vrf1
rd 100:1
route-target export 100:1
route-target import 100:1
exit
end
!
! apply VRF to the interface facing CE
interface ethernet 3/1
ip vrf forwarding vrf1
ip address 190.1.1.2 255.255.0.0
!
! make the interface facing the MPLS network an MPLS interface
interface serial 2/0
mpls ip
ip address 191.171.151.1 255.255.0.0
!
! configure BGP protocol for MPLS network
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 191.171.151.2 remote-as 100
neighbor 191.171.151.2 update-source serial 2/0
no auto-summary
address-family vpnv4
neighbor 191.171.151.2 activate
neighbor 191.171.151.2 send-community both
exit-address-family
address-family ipv4 vrf vrf1
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
```

```
!
! configure VRF static route to reach CE network
ip route vrf vrf1 192.168.4.0 255.255.255.0 190.1.1.1
```

VRF Configuration on PE2

```
! configure VRF for host2
ip cef
ip vrf vrf1
rd 100:1
route-target export 100:1
route-target import 100:1
!
! apply VRF on CE-facing interface
interface fastethernet 0/0
ip vrf forwarding vrf1
ip address 193.1.1.2 255.255.255.0
!
! make MPLS network-facing interface an MPLS interface
interface serial 1/0
mpls ip
ip address 191.171.151.2 255.255.0.0
!
! configure BGP protocol for MPLS network
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 191.171.151.1 remote-as 100
neighbor 191.171.151.1 update-source serial 1/0
no auto-summary
address-family vpnv4
neighbor 191.171.151.1 activate
neighbor 191.171.151.1 send-community both
exit-address-family
address-family ipv4 vrf vrf1
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
!configure VRF static route to reach CE network
ip route vrf vrf1 192.168.4.0 255.255.255.0 193.1.1.1
```

Configuration on CE1

```
interface e0/1
ip address 190.1.1.1 255.255.255.0
interface e0/0
ip address 192.168.4.2 255.255.255.0
ip route 192.168.104.0 255.255.255.0 190.1.1.2
```

Configuration on CE2

```
interface e0/1
ip address 190.1.1.1 255.255.255.0
interface e0/0
ip address 192.168.4.2 255.255.255.0
ip route 192.168.4.0 255.255.255.0 193.1.1.2
```

Configure Firewall on PE1 and Apply on the VRF Interface

```
! configure ACL so that NET2 cannot access NET1
ip access-list extended 105
permit tcp any any fragment
```

```
permit udp any any fragment
deny tcp any any
deny udp any any
permit ip any any
!
! apply ACL to VRF interface on PE1
interface ethernet 3/1
ip access-group 105 out
!
! configure firewall rule
ip inspect name test tcp
!
! apply firewall rule on VRF interface
interface ethernet 3/1
 ip inspect test in
```

Check for VRF Firewall Sessions When Host on NET1 Tries to Telnet to Server on NET2

```
show ip inspect session vrf vrf1
Established Sessions
 Session 659CE534 (192.168.4.1:38772)=>(192.168.104.1:23) tcp SIS_OPEN
!
! checking for ACLs
show ip inspect session detail vrf vrf1 | include ACL 105
  Out SID 192.168.104.1[23:23]=>192.168.4.1[38772:38772] on ACL 105
(34 matches)
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
VRF-lite	*Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide* , *Release 12.2*
MPLS VPN	*Configuring a Basic MPLS VPN* , *Document ID 13733*
VRF Aware IPSec	• *VRF-Aware IPSec* feature module, Release 12.2(15)T • *Cisco IOS Security Configuration Guide* , Release 12.3 • *Cisco IOS Security Command Reference* , Release 12.3T
VRF management	*Cisco 12000/10720 Router Manager User's Guide* , *Release 3.2*

Related Topic	Document Title
NAT	• *NAT and Stateful Inspection of Cisco IOS Firewall* , White Paper • *Configuring Network Address Translation: Getting Started* --Document ID 13772

Standards

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

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

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

Feature Information for VRF-Aware Cisco Firewall

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

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

Table 3: Feature Information for VRF-Aware Cisco Firewall

Feature Name	Releases	Feature Information
VRF-Aware Cisco Firewall	12.3(14)T	VRF-aware Cisco firewall applies Cisco firewall functionality to VRF (Virtual Routing and Forwarding) interfaces when the firewall is configured on a service provider (SP) or large enterprise edge device. SPs can provide managed services to small and medium business markets. The VRF-aware Cisco firewall supports VRF-aware URL filtering and VRF-lite (also known as Multi-VRF CE). The following commands were introduced or modified:**clearipurlfiltercache**, **ipinspectalert-off**, **ipinspectaudittrail**, **ipinspectdns-timeout**, **ipinspectmax-incompletehigh**, **ipinspectmax-incompletelow**, **ipinspectname**, **ipinspectone-minutehigh**, **ipinspectone-minutelow**, **ipinspecttcpfinwait-time**, **ipinspecttcpidle-time**, **ipinspecttcpmax-incompletehost**, **ipinspecttcpsynwait-time**, **ipinspectudpidle-time**, **ipurlfilteralert**, **ipurlfilterallowmode**, **ipurlfilteraudit-trail**, **ipurlfiltercache**, **ipurlfilterexclusive-domain**, **ipurlfilterexclusive-domain**, **ipurlfiltermax-request**, **ipurlfiltermax-resp-pak**, **ipurlfilterservervendor**, **ipurlfilterurlf-server-log**, **showipinspect**, **showipurlfiltercache**, **showipurlfilterconfig**, **showipurlfilterstatistics**.

Glossary

CE router --customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router.

CBAC --Context-Based Access Control. A protocol that provides internal users with secure access control for each application and for all traffic across network perimeters. CBAC enhances security by scrutinizing both source and destination addresses and by tracking each application's connection status.

data authentication --Refers to one or both of the following: data integrity, which verifies that data has not been altered, or data origin authentication, which verifies that the data was actually sent by the claimed sender.

data confidentiality --A security service where the protected data cannot be observed.

edge router --A router that turns unlabeled packets into labeled packets, and vice versa.

firewall --A router or access server, or several routers or access servers, designated as a buffer between any connected public networks and a private network. A firewall router uses access lists and other methods to ensure the security of the private network.

inspection rule --A rule that specifies what IP traffic (which application-layer protocols) will be inspected by CBAC at an interface.

intrusion detection --The Cisco IOS Firewall's Intrusion Detection System (Cisco IOS IDS) identifies the most common attacks, using signatures to detect patterns of misuse in network traffic.

IPSec --IP Security Protocol. A framework of open standards developed by the Internet Engineering Task Force (IETF). IPSec provides security for transmission of sensitive data over unprotected networks such as the Internet.

managed security services --A comprehensive set of programs that enhance service providers' abilities to meet the growing demands of their enterprise customers. Services based on Cisco solutions include managed firewall, managed VPN (network based and premises based), and managed intrusion detection.

NAT --Network Address Translation. Translates a private IP address used inside the corporation to a public, routable address for use outside of the corporation, such as the Internet. NAT is considered a one-to-one mapping of addresses from private to public.

PE router --provider edge router. A router that is part of a service provider's network and is connected to a customer edge (CE) router.

skinny --Skinny Client Control Protocol (SCCP). A protocol that enables CBAC to inspect Skinny control packets that are exchanged between a Skinny client and the Call Manager (CM); CBAC then configures the router (also known as the Cisco IOS Firewall) to enable the Skinny data channels to traverse through the router.

traffic filtering --A capability that allows you to configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall.

traffic inspection --CBAC inspection of traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions (sessions that originated from within the protected internal network).

UDP -- User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols.

VPN --Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

vrf --A VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) router.

VRF table --A table that stores routing data for each VPN. The VRF table defines the VPN membership of a customer site attached to the network access server (NAS). Each VRF table comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, and guidelines and routing protocol parameters that control the information that is included in the routing table.

Note Refer to Internetworking Terms and Acronyms for terms not included in this glossary.

CHAPTER 4

Zone-Based Policy Firewall High Availability

The Zone-Based Policy Firewall High Availability feature enables you to configure pairs of devices to act as backup for each other. High availability can be configured to determine the active device based on a number of failover conditions. When a failover occurs, the standby device seamlessly takes over and starts forwarding traffic and maintaining a dynamic routing table. The Zone-Based Policy Firewall High Availability feature supports active/active high availability, active/standby high availability, and asymmetric routing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Zone-Based Policy Firewall High Availability

- Interfaces attached to a firewall must have the same redundant interface identifier (RII).
- The active and standby devices must have the same zone-based policy firewall configuration.

- The active and standby devices must run on an identical version of the Cisco software. The active and standby devices must be connected through a switch.
- For asymmetric routing traffic to pass, you must configure the pass action for the class-default class.
- If you configure a zone pair between two LAN interfaces, ensure that you configure the same redundancy group (RG) on both interfaces. The zone pair configuration is not supported if LAN interfaces belong to different RGs.

Restrictions for Zone-Based Policy Firewall High Availability

- Asymmetric routing is not supported on interfaces that are a part of a redundancy group (RG).
- Asymmetric routing should not be used for load sharing of WAN links because very high asymmetric routing traffic can cause performance degradation of devices.
- A Layer 2 interface that is converted to a Layer 3 interface by using the **no switchport** command should not be used as a redundancy control link or a data link.
- In an active/active redundancy scenario, there should not be any traffic flow between the interfaces that are part of different RGs. For traffic flow between interfaces, both the interfaces should be part of the same zone or of a different zone with pass action configured between the zones.
- Multiprotocol Label Switching (MPLS) is not supported on asymmetric routing.
- Layer 7 inspection is not HA-aware. If Layer 7 inspection is enabled and the active RG goes down, only Layer 4 sessions will be synchronized to the standby RG; Layer 7 sessions have to be reestablished with the server.
- Zone-based policy firewall supports only Layer 4 protocol inspection with redundancy.
- VRFs are not supported and cannot be configured under ZBFW High Availability data and control interfaces.
- Configuring zone-based policy firewall high availability with NAT and NAT high availability with zone-based policy firewalls is not recommended.

Information About Zone-Based Policy Firewall High Availability

Zone-Based Policy Firewall High Availability Overview

High availability enables network-wide protection by providing fast recovery from faults that may occur in any part of a network. High availability enables rapid recovery from disruptions to users and network applications.

The zone-based policy firewall supports active/active and active/standby high availability failover and asymmetric routing.

The active/active failover allows both devices involved in the failover to forward traffic simultaneously.

When active/standby high availability failover is configured, only one of the devices involved in the failover handles the traffic at one time, while the other device is in a standby mode, periodically synchronizing session information from the active device.

Asymmetric routing supports the forwarding of packets from a standby redundancy group to an active redundancy group for packet handling. If this feature is not enabled, the return TCP packets forwarded to the device that did not receive the initial synchronization (SYN) message are dropped because they do not belong to any known existing session.

Zone-Based Policy Firewall High Availability Operation

You can configure pairs of devices to act as hot standby devices for each other. Redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups (RGs). An RG must be configured under the interface in order for the zone-based policy firewall to correctly replicate connections in a high availability setup. In order for the firewall to synchronize connections, an RG must be associated with an interface.

Figure 1 depicts an active/standby load-sharing scenario. It shows how a redundancy group is configured for a pair of devices that has one outgoing interface. Figure 2 depicts an active/active load-sharing scenario. It shows how two redundancy groups are configured for a pair of devices that have two outgoing interfaces.

In both cases, the redundant devices are joined by a configurable control link, a data synchronization link, and an interlink interface. The control link is used to communicate the status of the devices. The data synchronization link is used to transfer stateful information from the firewall and to synchronize the stateful database. The pairs of redundant interfaces are configured with the same unique ID number, known as the redundant interface identifier (RII).

Asymmetric routing is supported as part of the firewall high availability. In a LAN-WAN scenario, where the return traffic enters standby devices, asymmetric routing is supported. To implement the asymmetric routing functionality, configure both the redundant devices with a dedicated interface (interlink interface) for asymmetric traffic. This dedicated interface will redirect the traffic coming to the standby WAN interface to the active device.

Figure 10: Redundancy Group—One Outgoing Interface

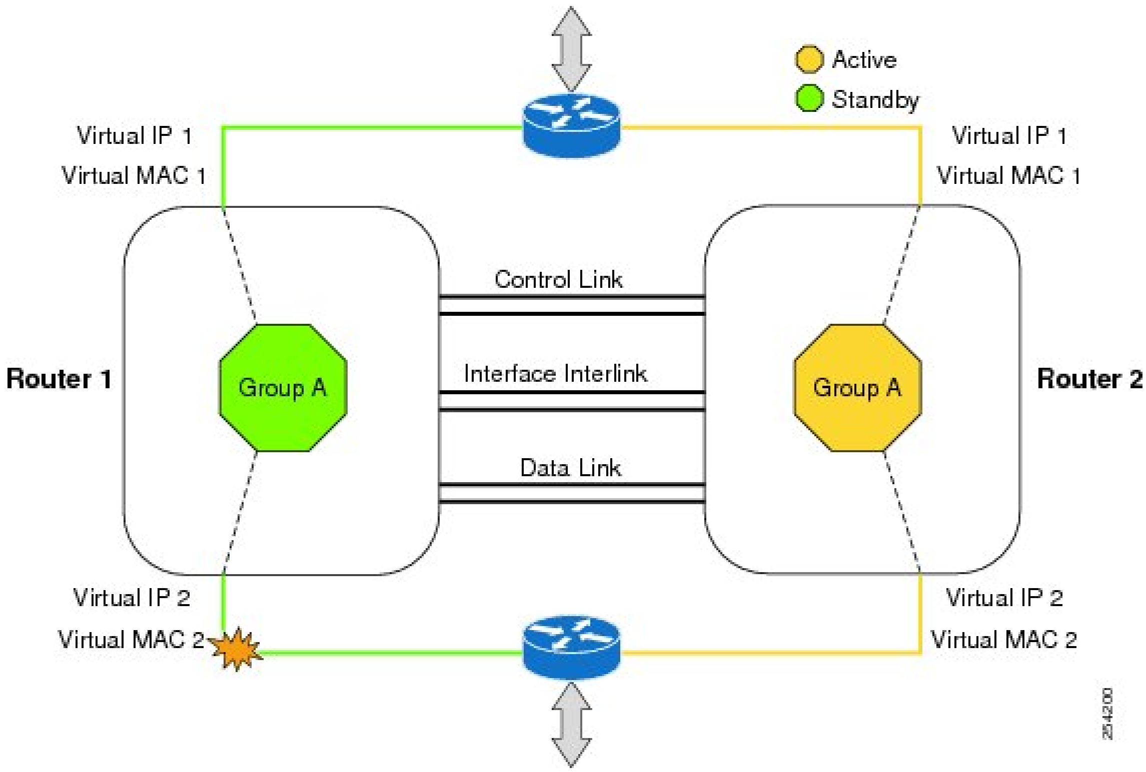
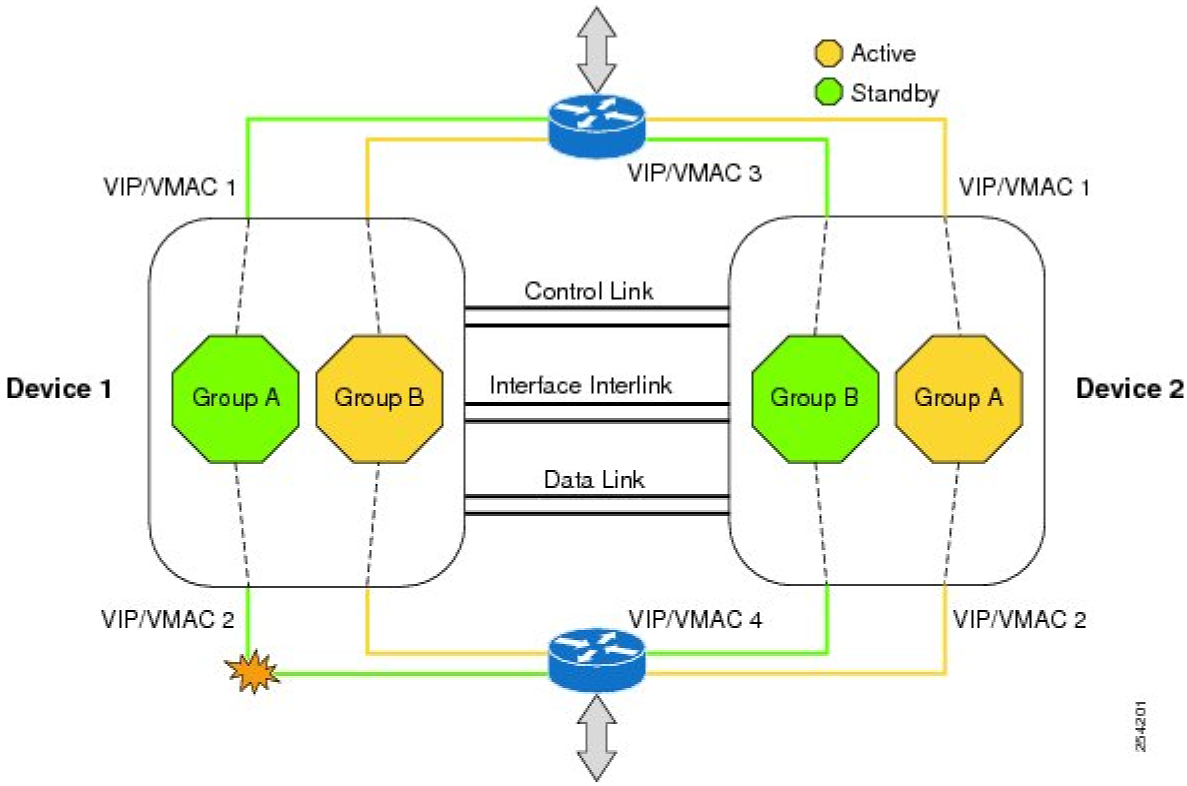


Figure 11: Redundancy Group Configuration—Two Outgoing Interfaces



The status of redundancy group members is determined through the use of hello messages sent over the control link. If either of the devices do not respond to a hello message within a configurable amount of time, the software considers that a failure has occurred, and a switchover is initiated. To detect a failure in milliseconds, the control links run the failover protocol. You can configure the following parameters for hello messages:

- Active timer.
- Standby timer.
- Hello time—The interval at which hello messages are sent.
- Hold time—The amount of time before which the active or standby device is declared to be down.

The hello time defaults to 3 seconds to align with the Hot Standby Router Protocol (HSRP), and the hold time defaults to 10 seconds. You can also configure these timers in milliseconds by using the **timers hellotime msec** command.

To determine which pairs of interfaces are affected by the switchover, you must configure a unique ID for each pair of redundant interfaces. This ID is known as the RII that is associated with the interface.

A switchover to the standby device can occur under other circumstances. Another factor that can cause a switchover is a priority setting that can be configured on each device. The device with the highest priority value will be the active device. If a fault occurs on either the active or the standby device, the priority of the device is decremented by a configurable amount, known as the weight. If the priority of the active device falls below the priority of the standby device, a switchover occurs and the standby device becomes the active device. This default behavior can be overridden by disabling the preemption attribute for the redundancy group. You can also configure each interface to decrease the priority when the Layer 1 state of the interface goes down. The priority that is configured overrides the default priority of a redundancy group.

Each failure event that causes a modification of a redundancy group's priority generates a syslog entry that contains a time stamp, the redundancy group that was affected, the previous priority, the new priority, and a description of the failure event cause.

Another situation that can cause a switchover to occur is when the priority of a device or interface falls below a configurable threshold level.

A switchover to the standby device occurs under the following circumstances:

- Power loss or a reload occurs on the active device (this includes crashes).
- The run-time priority of the active device goes down below that of the standby device.
- The run-time priority of the active device goes down below the configured threshold device.
- The redundancy group on the active device is reloaded manually by using the **redundancy application reload group** *rg-number* command.
- Two consecutive hello messages missed on any monitored interface forces the interface into testing mode. Both devices will verify the link status on the interface and then execute the following tests:
    - Network activity test
    - Address Resolution Protocol (ARP) test
    - Broadcast ping test

Active/Active Failover

In an active/active failover configuration, both devices can process network traffic. Active/active failover generates virtual MAC (VMAC) addresses for interfaces in each redundancy group (RG).

One device in an active/active failover pair is designated as the primary (active) device, and the other is designated as the secondary (standby) device. Unlike with active/standby failover, this designation does not indicate which device becomes active when both devices start simultaneously. Instead, the primary/secondary designation determines the following:

- The device that provides the running configuration to the failover pair when they start simultaneously.
- The device on which the failover RG appears in the active state when devices start simultaneously. Each failover RG in the configuration is configured with a primary or secondary device preference. You can configure both failover RGs to be in the active state on a single device and the standby failover RGs to be on the other device. You can also configure one failover RG to be in the active state and the other RG to be in the standby state on a single device.

Active/Standby Failover

Active/standby failover enables you to use a standby device to take over the functionality of a failed device. A failed active device changes to the standby state, and the standby device changes to the active state. The device that is now in the active state takes over IP addresses and MAC addresses of the failed device and starts processing traffic. The device that is now in the standby state takes over standby IP addresses and MAC addresses. Because network devices do not see any change in the MAC-to-IP address pairing, Address Resolution Protocol (ARP) entries do not change or time out anywhere on the network.

In an active/standby scenario, the main difference between two devices in a failover pair depends on which device is active and which device is a standby, namely which IP addresses to use and which device actively passes the traffic. The active device always becomes the active device if both devices start up at the same time (and are of equal operational health). MAC addresses of the active device are always paired with active IP addresses.

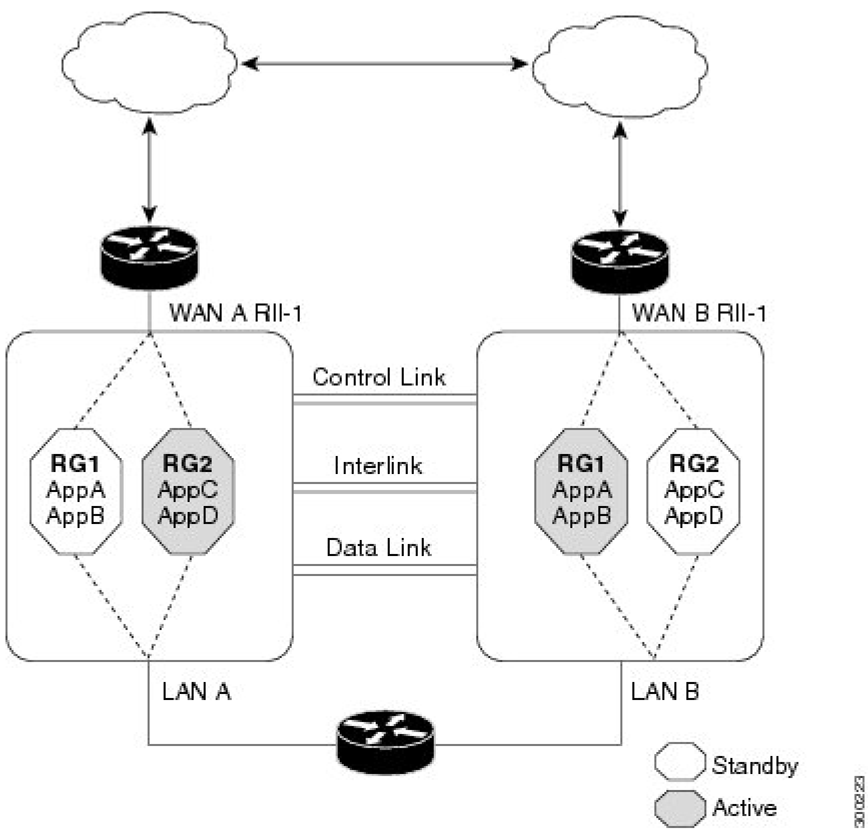
Asymmetric Routing Overview

Asymmetric routing occurs when packets from TCP or UDP connections flow in different directions through different routes. In asymmetric routing, packets that belong to a single TCP or UDP connection are forwarded through one interface in a redundancy group (RG), but returned through another interface in the same RG. In asymmetric routing, the packet flow remains in the same RG. When you configure asymmetric routing, packets received on the standby RG are redirected to the active RG for processing. If asymmetric routing is not configured, the packets received on the standby RG may be dropped.

Asymmetric routing determines the RG for a particular traffic flow. The state of the RG is critical in determining the handling of packets. If an RG is active, normal packet processing is performed. In case the RG is in a standby state and you have configured asymmetric routing and the **asymmetric-routing always-divert enable** command, packets are diverted to the active RG. Use the **asymmetric-routing always-divert enable** command to always divert packets received from the standby RG to the active RG.

The figure below shows an asymmetric routing scenario with a separate asymmetric-routing interlink interface to divert packets to the active RG.

Figure 12: Asymmetric Routing Scenario



The following rules apply to asymmetric routing:

- 1:1 mapping exists between the redundancy interface identifier (RII) and the interface.
- 1:*n* mapping exists between the interface and an RG. (An asymmetric routing interface can receive traffic from and send traffic to multiple RGs. For a non asymmetric-routing interface (normal LAN interface), a 1:1 mapping exists between the interface and the RG.)
- 1:*n* mapping exists between an RG and applications that use it. (Multiple applications can use the same RG).
- 1:1 mapping exists between an RG and the traffic flow. The traffic flow must map only to a single RG. If a traffic flow maps to multiple RGs, an error occurs.
- 1:1 or 1:*n* mapping can exist between an RG and an asymmetric-routing interlink as long as the interlink has sufficient bandwidth to support all the RG interlink traffic.

Asymmetric routing consists of an interlink interface that handles all traffic that is to be diverted. The bandwidth of the asymmetric-routing interlink interface must be large enough to handle all expected traffic that is to be diverted. An IPv4 address must be configured on the asymmetric-routing interlink interface, and the IP address of the asymmetric routing interface must be reachable from this interface.

Note We recommend that the asymmetric-routing interlink interface be used for interlink traffic only and not be shared with high availability control or data interfaces because the amount of traffic on the asymmetric-routing interlink interface could be quite high.

WAN-LAN Topology

In a WAN-LAN topology, two devices are connected through LAN interfaces on the inside and WAN interfaces on the outside. There is no control on the routing of return traffic received through WAN links.

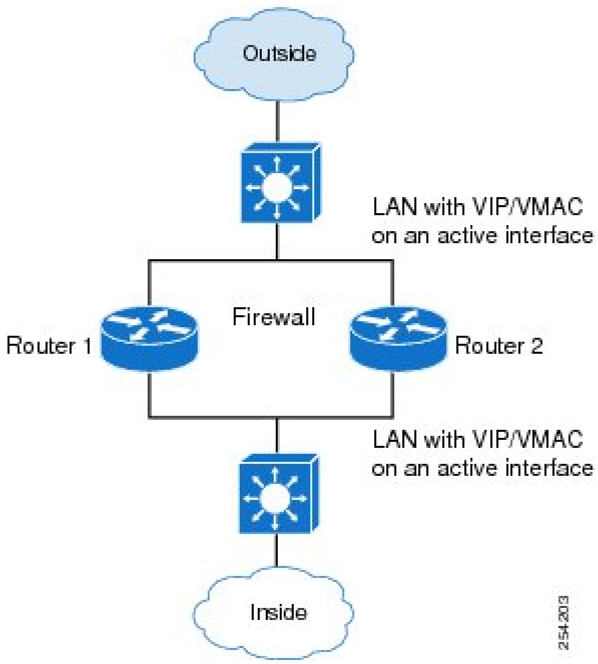
WAN links can be provided by the same service provider or different service providers. In most cases, WAN links are provided by different service providers. To utilize WAN links to the maximum, configure an external device to provide a failover.

On LAN-based interfaces, a high availability virtual IP address is required to exchange client information and for faster failover. On WAN-based interfaces, the **redundancy group** *id* **ip** *virtual-ip* **decrement** *value* command is used for failover.

LAN-LAN Topology

In a LAN-LAN topology, all participating devices are connected to each other through LAN interfaces on both the inside and the outside. In this scenario, the traffic is often directed to the correct firewall if static routing is configured on the upstream or downstream devices to an appropriate virtual IP address. The dynamic routing configuration supported on LAN-facing interfaces must not introduce a dependency on routing protocol convergence; otherwise, fast failover requirements will not be met. The figure below shows a LAN-LAN topology.

Figure 13: LAN-LAN Scenario



Exclusive Virtual IP Addresses and Exclusive Virtual MAC Addresses

Virtual IP (VIP) addresses and virtual MAC (VMAC) addresses are used by security applications to control interfaces that receive traffic. An interface is paired with another interface, and these interfaces are associated with the same redundancy group (RG). The interface that is associated with an active RG exclusively owns the VIP and VMAC. The Address Resolution Protocol (ARP) process on the active device sends ARP replies for any ARP request for the VIP, and the Ethernet controller for the interface is programmed to receive packets destined for the VMAC. When an RG failover occurs, the ownership of the VIP and VMAC changes. The interface that is associated with the newly active RG sends a gratuitous ARP and programs the interface's Ethernet controller to accept packets destined for the VMAC.

IPv6 Support

You can assign each redundancy group (RG) on a traffic interface for both IPv4 and IPv6 virtual IP (VIP) addresses under the same redundancy interface identifier (RII). Each RG uses a unique virtual MAC (VMAC) address per RII. For an RG, the IPv6 link-local VIP and global VIP coexist on an interface.

You can configure an IPv4 VIP, a link-local IPv6 VIP, and/or a global IPv6 VIP for each RG on a traffic interface. IPv6 link-local VIP is mainly used when configuring static or default routes, whereas IPv6 global VIP is widely used in both LAN and WAN topologies.

You must configure a physical IP address before configuring an IPv4 VIP.

Virtual Fragmentation Reassembly

Virtual fragmentation reassembly (VFR) enables the firewall to create dynamic access control lists (ACLs) to protect the network from various fragmentation attacks. VFR is high availability-aware. When the firewall is enabled for high availability, fragmented packets that arrive on the standby redundancy group (RG) are redirected to the active redundancy group. Use the **ip virtual-reassembly** command to enable VFR on an interface.

Note VFR should not be enabled on a device that is placed on an asymmetric path. The reassembly process requires all fragments within an IP datagram. Devices placed in the asymmetric path may not receive all IP fragments, and the fragment reassembly will fail.

How to Configure Zone-Based Policy Firewall High Availability

Configuring Application Redundancy and Redundancy Application Groups

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **redundancy**
5. **log dropped-packets enable**
6. **exit**
7. **redundancy**
8. **application redundancy**
9. **group** *id*
10. **name** *group-name*
11. **preempt**
12. **priority** *value*
13. **control** *interface-type interface-number* **protocol** *id*
14. **data** *interface-type interface-number*
15. **asymmetric-routing interface** *type number*
16. Configure Step 7 to Step 11 to create another redundancy group on the same device.
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Defines a global inspect parameter map and enters parameter-map type inspect configuration mode.
Step 4	**redundancy** **Example:** `Device(config-profile)# redundancy`	Enables firewall high availability.
Step 5	**log dropped-packets enable** **Example:** `Device(config-profile)# log dropped-packets enable`	Enables logging of packets dropped by the firewall.
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and returns to global configuration mode.
Step 7	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 8	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 9	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a group and enters redundancy application group configuration mode.

	Command or Action	Purpose
Step 10	**name** *group-name* **Example:** `Device(config-red-app-grp)# name RG1`	Configures a redundancy group with a name.
Step 11	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the redundancy group.
Step 12	**priority** *value* **Example:** `Device(config-red-app-grp)# priority 230`	Specifies a group priority and a failover threshold value for a redundancy group.
Step 13	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1`	Configures the control interface type and number for a redundancy group.
Step 14	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data gigabitethernet 0/0/1`	Configures the data interface type and number for a redundancy group.
Step 15	**asymmetric-routing interface** *type number* **Example:** `Device(config-red-app-grp)# asymmetric-routing interface gigabitethernet 0/0/1`	Enables asymmetric routing on an interface.
Step 16	Configure Step 7 to Step 11 to create another redundancy group on the same device.	—
Step 17	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and returns to privileged EXEC mode.

Configuring a Firewall for High Availability

In this task, you will do the following:

- Configure a firewall.
- Create a security source zone.

- Create a security destination zone.
- Create a security zone pair by using the configured source and destination zones.
- Configure an interface as a zone member.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **parameter-map type inspect global**
7. **redundancy**
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** *class-map-name*
11. **inspect**
12. **exit**
13. **class class-default**
14. **drop**
15. **exit**
16. **exit**
17. **zone security** *zone-name*
18. **exit**
19. **zone security** *zone-name*
20. **exit**
21. **zone-pair security** *zone-pair-name* **source** *zone-name* **destination** *zone-name*
22. **service-policy type inspect** *policy-map-name*
23. **exit**
24. **zone-pair security** *zone-pair-name* **source** *zone-name* **destination** *zone-name*
25. **service-policy type inspect** *policy-map-name*
26. **exit**
27. **interface** *type number*
28. **ip address** *ip-address mask*
29. **encapsulation dot1q** *vlan-id*
30. **zone-member security** *security-zone-name*
31. **end**
32. **show policy-firewall session zone-pair ha**
33. **debug policy-firewall ha**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any cmap-l4-Protocol`	Defines the class on which an action is to be performed and enters policy-map class configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol tcp`	Configures a match criterion for a class map on the basis of the specified protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits policy-map class configuration mode and returns to global configuration mode.
Step 6	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Defines a global inspect parameter map and enters parameter-map type inspect configuration mode.
Step 7	**redundancy** **Example:** `Device(config-profile)# redundancy`	Enables firewall high availability.
Step 8	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and returns to global configuration mode.
Step 9	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect pmap-l4-Protocols`	Creates a protocol-specific inspect type policy map and enters policy-map configuration mode.

	Command or Action	Purpose
Step 10	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect cmap-l4-Protocol`	Defines the class on which an action is to be performed and enters policy-map class configuration mode.
Step 11	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 13	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Configures the default class on which an action is to be performed and enters policy-map class configuration mode.
Step 14	**drop** **Example:** `Device(config-pmap-c)# drop`	Drops packets that are sent to a device.
Step 15	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 16	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode and returns to global configuration mode.
Step 17	**zone security** *zone-name* **Example:** `Device(config)# zone security TWAN`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair: a source and a destination zone.
Step 18	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 19	**zone security** *zone-name* **Example:** `Device(config)# zone security DATA`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair: a source and a destination zone.
Step 20	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 21	**zone-pair security** *zone-pair-name* **source** *zone-name* **destination** *zone-name* **Example:** `Device(config)# zone-pair security zp-TWAN-DATA source TWAN destination data`	Creates a zone pair to which interfaces can be assigned and enters security zone-pair configuration mode.
Step 22	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect pmap-l4-Protocols`	Attaches a firewall policy map to a zone pair.
Step 23	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 24	**zone-pair security** *zone-pair-name* **source** *zone-name* **destination** *zone-name* **Example:** `Device(config)# zone-pair security zp-DATA-TWAN source DATA destination TWAN`	Creates a zone pair to which interfaces can be assigned and enters security zone-pair configuration mode.
Step 25	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect pmap-l4-Protocols`	Attaches a firewall policy map to a zone pair.
Step 26	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone pair configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 27	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an IP address for the subinterface.
Step 28	**ip address** *ip-address mask* **Example:** `Device(config-subif)# ip address 10.1.1.1 255.255.255.0`	Configures an IP address for the subinterface.
Step 29	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 2`	Sets the encapsulation method used by the interface.
Step 30	**zone-member security** *security-zone-name* **Example:** `Device(config-subif)# zone-member security private`	Configures the interface as a zone member. • For the *security-zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the device or initiated by the device) is dropped by default. To permit traffic through an interface that is a zone member, you must make that zone part of a zone pair to which you apply a policy. If the policy permits traffic (via **inspect** or **inspect** actions), traffic can flow through the interface.
Step 31	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone pair configuration mode and returns to privileged EXEC mode.
Step 32	**show policy-firewall session zone-pair ha** **Example:** `Device# show policy-firewall session zone-pair ha`	(Optional) Displays the firewall HA sessions pertaining to a zone pair.
Step 33	**debug policy-firewall ha** **Example:** `Device# debug policy-firewall ha`	(Optional) Displays messages about firewall events.

Configuring a Redundancy Application Group on a WAN Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **description** *string*
5. **ip address** *ip-address mask*
6. **zone-member security** *zone-name*
7. **ip tcp adjust-mss** *max-segment-size*
8. **redundancy rii** *RII-identifier*
9. **redundancy asymmetric-routing enable**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/2.1`	Configures a subinterface and enters subinterface configuration mode.
Step 4	**description** *string* **Example:** `Device(config-subif)# description wan interface`	Adds a description to an interface configuration.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-subif)# ip address 10.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

	Command or Action	Purpose
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-subif)# zone-member security TWAN`	Configures the interface as a zone member while configuring a firewall. • For the *zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the router or initiated by the router) is dropped by default. To permit traffic through an interface that is a zone member, you must make that zone part of a zone pair to which you apply a policy. If the policy permits traffic (via inspect or pass actions), traffic can flow through the interface.
Step 7	**ip tcp adjust-mss** *max-segment-size* **Example:** `Device(config-subif)# ip tcp adjust-mss 1360`	Adjusts the maximum segment size (MSS) value of TCP SYN packets going through a router.
Step 8	**redundancy rii** *RII-identifier* **Example:** `Device(config-subif)# redundancy rii 360`	Configures an RII for redundancy group-protected traffic interfaces.
Step 9	**redundancy asymmetric-routing enable** **Example:** `Device(config-subif)# redundancy asymmetric-routing enable`	Associates a redundancy group with an interface that is used for asymmetric routing.
Step 10	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.

Configuring a Redundancy Application Group on a LAN Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **description** *string*
5. **encapsulation dot1q** *vlan-id*
6. **ip vrf forwarding** *name*
7. **ip address** *ip-address mask*
8. **zone-member security** *zone-name*
9. **redundancy rii** *RII-identifier*
10. **redundancy group** *id* **ip** *ip-address* **exclusive**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/2.1`	Configures a subinterface and enters subinterface configuration mode.
Step 4	**description** *string* **Example:** `Device(config-subif)# description lan interface`	Adds a description to an interface configuration.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 18`	Sets the encapsulation method used by the interface.

	Command or Action	Purpose
Step 6	**ip vrf forwarding** *name* **Example:** `Device(config-subif)# ip vrf forwarding trust`	Associates a VPN routing and forwarding (VRF) instance with an interface or subinterface. • The command will not be configured if the specified VRF is not configured.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-subif)# ip address 10.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 8	**zone-member security** *zone-name* **Example:** `Device(config-subif)# zone-member security data`	Configures the interface as a zone member. • For the *zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command while configuring a firewall. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the router or initiated by the router) is dropped by default. To permit traffic through an interface that is a zone member, you must make that zone part of a zone pair to which you apply a policy. If the policy permits traffic (via inspect or pass actions), traffic can flow through the interface.
Step 9	**redundancy rii** *RII-identifier* **Example:** `Device(config-subif)# redundancy rii 100`	Configures an RII for redundancy group-protected traffic interfaces.
Step 10	**redundancy group** *id* **ip** *ip-address* **exclusive** **Example:** `Device(config-subif)# redundancy group 1 ip 10.0.0.1 exclusive`	Configures a virtual IP address for the redundancy group.
Step 11	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.

Configuration Examples for Zone-Based Policy Firewall High Availability

Example: Configuring Application Redundancy and Redundancy Application Groups

```
configure terminal
 parameter-map type inspect global
  redundancy
  log dropped-packets enable
!
  redundancy
   application redundancy
    group 1
     name RG1
     preempt
     priority 230
     control gigabitethernet 0/0/1 protocol 1
     data gigabitethernet 0/0/1
     asymmetric-routing gigabitethernet 0/0/1
```

Example: Configuring a Firewall for High Availability

```
configure terminal
 class-map type inspect match-any cmap-l4-Protocol
  match protocol tcp
!
 parameter-map type inspect global
  redundancy
!
 policy-map type inspect pmap-l4-Protocols
  class type inspect cmap-l4-Protocol
   inspect
!
  class class-default
   drop
!
!
 zone security TWAN
!
 zone security DATA
!
 zone-pair security zp-TWAN-DATA source TWAN destination DATA
  service-policy type inspect pmap-l4-Protocols
!
 zone-pair security zp-DATA-TWAN source DATA destination TWAN
  service-policy type inspect pmap-l4-Protocols
!
 interface gigabitethernet 0/0/0
  ip address 10.1.1.1 255.255.255.0
  encapsulation dot1q 2
  zone member security private
```

Example: Configuring a Redundancy Application Group on a WAN Interface

The following example shows how to configure redundancy groups for a WAN-LAN scenario:

```
interface gigabitethernet 0/0/2
 description wan interface
 ip 10.0.0.1 255.255.255.0
 zone-member security TWAN
 ip tcp adjust-mss 1360
 redundancy rii 360
 redundancy asymmetric-routing enable
```

The following is a sample WAN–LAN active/active configuration in which two devices have two LAN interfaces and one WAN interface. Two redundancy groups (RG1 and RG2) are configured on each device, and LAN interfaces are bound to one redundancy group. The WAN link is shared by both the RGs. RG1 is active on Device 1 and RG2 is active on Device 2.

```
! Configuration on Device 1:
redundancy
 application
  group 1
   name RG1
   priority 205 failover-threshold 200
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
   asymmetric-routing gigabitethernet 0/0/3
  group 2
   name RG2
   priority 195 failover-threshold 190
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
   asymmetric-routing gigabitethernet 0/0/3
!
!
parameter-map type inspect global
 redundancy
 redundancy delay 10
!
class-map type inspect match-all ha-class
 match protocol tcp
!
!
policy-map type inspect ha-policy
 class type inspect ha-class
  inspect
 class class-default
  drop
!
zone security ha-in
!
zone security ha-out
!
zone-pair security ha-in-out source ha-in destination ha-out
 service-policy type inspect ha-policy
!
!
interface pos 2/1
 redundancy rii 210 decrement 100
 redundancy asymmetric-routing enable
 zone-member security ha-out
!
interface gigabitethernet 0/0
 redundancy rii 1
 redundancy 1 ip 10.1.1.254 exclusive decrement 50
 zone-member security ha-in
!
!
interface gigabitethernet 0/1
 redundancy rii 2
```

```
 redundancy 1 ip 192.168.7.2 exclusive decrement 50
 zone-member security ha-in
!
! Configuration on Device 2:
redundancy
 application
  group 1
   name RG1
   priority 195 failover-threshold 190
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
   asymmetric-routing gigabitethernet 0/0/3
  group 2
   name RG2
   priority 205 failover-threshold 200
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
   asymmetric-routing gigabitethernet 0/0/3
!
!
parameter-map type inspect global
 redundancy
 redundancy delay 10
!
class-map type inspect match-all ha-class
 match protocol tcp
!
!
policy-map type inspect ha-policy
 class type inspect ha-class
  inspect
 class class-default
  drop
!
zone security ha-in
!
zone security ha-out
!
zone-pair security ha-in-out source ha-in destination ha-out
 service-policy type inspect ha-policy
!
!
interface pos 2/1
 redundancy rii 210 decrement 100
 redundancy asymmetric-routing enable
 zone-member security ha-out
!
interface gigabitethernet 0/0
 redundancy rii 1
 redundancy 1 ip 10.1.1.254 exclusive decrement 50
 zone-member security ha-in
!
!
interface gigabitethernet 0/1
 redundancy rii 2
 redundancy 2 ip 192.168.7.2 exclusive decrement 50
 zone-member security ha-in
```

The following is a sample active/standby LAN-WAN configuration with one LAN interface and one WAN interface on each device. Only one redundancy group (RG1) is configured, and it is active on Device 1 and on the standby on Device 2. The VIP address is owned by the LAN interface of the active device.

```
! Configuration on Device 1 (active):
redundancy
 application
  group 1
   name RG1
   priority 205 failover-threshold 200
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
   asymmetric-routing gigabitethernet 0/0/3
!
!
```

```
parameter-map type inspect global
 redundancy
 redundancy delay 10
!
class-map type inspect match-all ha-class
 match protocol tcp
!
!
policy-map type inspect ha-policy
 class type inspect ha-class
  inspect
 class class-default
  drop
!
zone security ha-in
!
zone security ha-out
!
zone-pair security ha-in-out source ha-in destination ha-out
 service-policy type inspect ha-policy
!
!
interface pos 2/1
 redundancy rii 210 decrement 100
 redundancy asymmetric-routing enable
 zone-member security ha-out
!
interface gigabitethernet 0/0
 redundancy rii 1
 redundancy 1 ip 10.1.1.254 exclusive decrement 50
 zone-member security ha-in


! Configuration on Device 2(standby):
redundancy
 application
  group 1
   name RG1
   priority 195 failover-threshold 190
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
   asymmetric-routing gigabitethernet 0/0/3
!
!
parameter-map type inspect global
 redundancy
 redundancy delay 10
!
class-map type inspect match-all ha-class
 match protocol tcp
!
!
policy-map type inspect ha-policy
 class type inspect ha-class
  inspect
 class class-default
  drop
!
zone security ha-in
!
zone security ha-out
!
zone-pair security ha-in-out source ha-in destination ha-out
 service-policy type inspect ha-policy
!
!
interface pos 2/1
 redundancy rii 210 decrement 100
 redundancy asymmetric-routing enable
 zone-member security ha-out
!
interface gigabitethernet 0/0
 redundancy rii 1
```

```
 redundancy 1 ip 10.1.1.254 exclusive decrement 50
 zone-member security ha-in
```

Example: Configuring a Redundancy Application Group on a LAN Interface

```
interface gigabitethernet 0/0/2
 description lan interface
 ip address 10.0.0.1 255.255.255.0
 zone member security data
 redundancy rii 100
 redundancy group 1 ip 10.0.0.1 exclusive
```

The following is an active/active LAN-LAN configuration that has a device with two LAN interfaces for both upstream and downstream traffic. Two redundancy groups (RG1 and RG2) are configured on each device. The pairing for each LAN upstream and LAN downstream links exists, and each pair is made part of a single redundancy group. In this scenario, the VIP addresses and VMAC address ownership is exclusively restricted to the active interface and hence there is no possibility of asymmetric routing.

```
! Configuration on Device 1:
redundancy
 application
  group 1
   name RG1
   priority 205 failover-threshold 200
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
  group 2
   name RG2
   priority 195 failover-threshold 190
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
!
!
parameter-map type inspect global
 redundancy
 redundancy delay 10
!
class-map type inspect match-all ha-class
 match protocol tcp
!
!
policy-map type inspect ha-policy
 class type inspect ha-class
  inspect
 class class-default
  drop
!
zone security ha-in
!
zone security ha-out
!
zone-pair security ha-in-out source ha-in destination ha-out
 service-policy type inspect ha-policy
!
!
interface gigabitethernet 0/0
 redundancy rii 1
 redundancy 1 ip 10.1.1.254 exclusive decrement 50
 zone-member security ha-in
!
!
interface gigabitethernet 0/1
 redundancy rii 2
 redundancy 2 ip 10.3.1.254 exclusive decrement 50
 zone-member security ha-in
!
!
interface gigabitethernet 1/0
 redundancy rii 210 decrement 100
```

```
 redundancy 1 ip 10.2.1.254 exclusive decrement 50
 zone-member security ha-out
!
interface gigabitethernet 1/1
 redundancy rii 110 decrement 100
 redundancy 2 ip 10.4.1.254 exclusive decrement 50
 zone-member security ha-out
!
! Configuration on Device 2:
redundancy
 application
  group 1
   name RG1
   priority 195 failover-threshold 190
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
  group 2
   name RG2
   priority 205 failover-threshold 200
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
!
!
parameter-map type inspect global
 redundancy
 redundancy delay 10
!
class-map type inspect match-all ha-class
 match protocol tcp
!
!
policy-map type inspect ha-policy
 class type inspect ha-class
  inspect
 class class-default
  drop
!
zone security ha-in
!
zone security ha-out
!
zone-pair security ha-in-out source ha-in destination ha-out
 service-policy type inspect ha-policy
!
!
interface gigabitethernet 0/0
 redundancy rii 1
 redundancy 1 ip 10.1.1.254 exclusive decrement 50
 zone-member security ha-in
!
!
interface gigabitethernet 0/1
 redundancy rii 2
 redundancy 2 ip 10.3.1.254 exclusive decrement 50
 zone-member security ha-in
!
!
interface gigabitethernet 1/0
 redundancy rii 210 decrement 100
 redundancy 1 ip 10.2.1.254 exclusive decrement 50
 zone-member security ha-out
!
interface gigabitethernet 1/1
 redundancy rii 110 decrement 100
 redundancy 2 ip 10.4.1.254 exclusive decrement 50
 zone-member security ha-out
```

The following is an active/standby LAN-LAN configuration. This configuration is similar to the active/standby WAN-LAN configuration in which each device has one LAN interface for both upstream and downstream traffic. Only one redundancy group (RG1) is configured and each interface is made part of this redundancy group.

```
! Configuration on Device 1 (active):
redundancy
 application
  group 1
   name RG1
   priority 205 failover-threshold 200
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
!
!
parameter-map type inspect global
 redundancy
 redundancy delay 10
!
class-map type inspect match-all ha-class
 match protocol tcp
!
!
policy-map type inspect ha-policy
 class type inspect ha-class
  inspect
 class class-default
  drop
!
zone security ha-in
!
zone security ha-out
!
zone-pair security ha-in-out source ha-in destination ha-out
 service-policy type inspect ha-policy
!
!
interface gigabitethernet 0/0
 redundancy rii 1
 redundancy 1 ip 10.1.1.254 exclusive decrement 50
 zone-member security ha-out
!
!
interface gigabitethernet 1/0
 redundancy rii 210 decrement 100
 redundancy 1 ip 10.2.1.254 exclusive decrement 50
 zone-member security ha-out
!
! Configuration on Device 2(standby):
redundancy
 application
  group 1
   name RG1
   priority 195 failover-threshold 190
   control gigabitethernet 0/0/1 protocol 1
   data gigabitethernet 0/0/2
!
!
parameter-map type inspect global
 redundancy
 redundancy delay 10
!
class-map type inspect match-all ha-class
 match protocol tcp
!
!
policy-map type inspect ha-policy
 class type inspect ha-class
  inspect
 class class-default
  drop
!
zone security ha-in
!
```

```
zone security ha-out
!
zone-pair security ha-in-out source ha-in destination ha-out
 service-policy type inspect ha-policy
!
!
interface gigabitethernet 0/0
 redundancy rii 1
 redundancy 1 ip 10.1.1.254 exclusive decrement 50
 zone-member security ha-out
!
!
interface gigabitethernet 1/0
 redundancy rii 210 decrement 100
 redundancy 1 ip 10.2.1.254 exclusive decrement 50
 zone-member security ha-out
```

Feature Information for Zone-Based Policy Firewall High Availability

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

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

Table 4: Feature Information for Zone-Based Policy Firewall High Availability

Feature Name	Releases	Feature Information
Zone-Based Policy Firewall High Availability	15.2(3)T	The Zone-Based Policy Firewall High Availability feature enables you to configure pairs of routers to act as backup for each other. High availability (HA) can be configured to determine the active router based on a number of failover conditions. When a failover occurs, the standby router seamlessly takes over and starts forwarding traffic and maintaining a dynamic routing table. The Zone-Based Policy Firewall High Availability feature supports active/active HA, active/standby HA, and asymmetric routing. The following commands were introduced or modified: **debug policy-firewall**, **redundancy**, and **show policy-firewall**.

CHAPTER 5

Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls

The Interchassis Asymmetric Routing Support for Zone-Based Firewalls feature supports the forwarding of packets from a standby redundancy group to an active redundancy group for packet handling. If this feature is not enabled, the return TCP packets forwarded to the router that did not receive the initial synchronization (SYN) message are dropped because they do not belong to any known existing session. Interchassis asymmetric routing also supports active/active and active/standby load sharing redundancy.

This module provides an overview of asymmetric routing and active/active and active/standby load sharing redundancy, and describes how to configure asymmetric routing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Restrictions for Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls

The following are not supported:

- Asymmetric routing on a Multiprotocol Label Switching (MPLS) VPN network. You cannot configure MPLS on the egress interface and VPN routing and forwarding (VRF) on the ingress interface.
- Configuring asymmetric routing on a redundancy group (RG) interface.
- IPv6 traffic.

Information About Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls

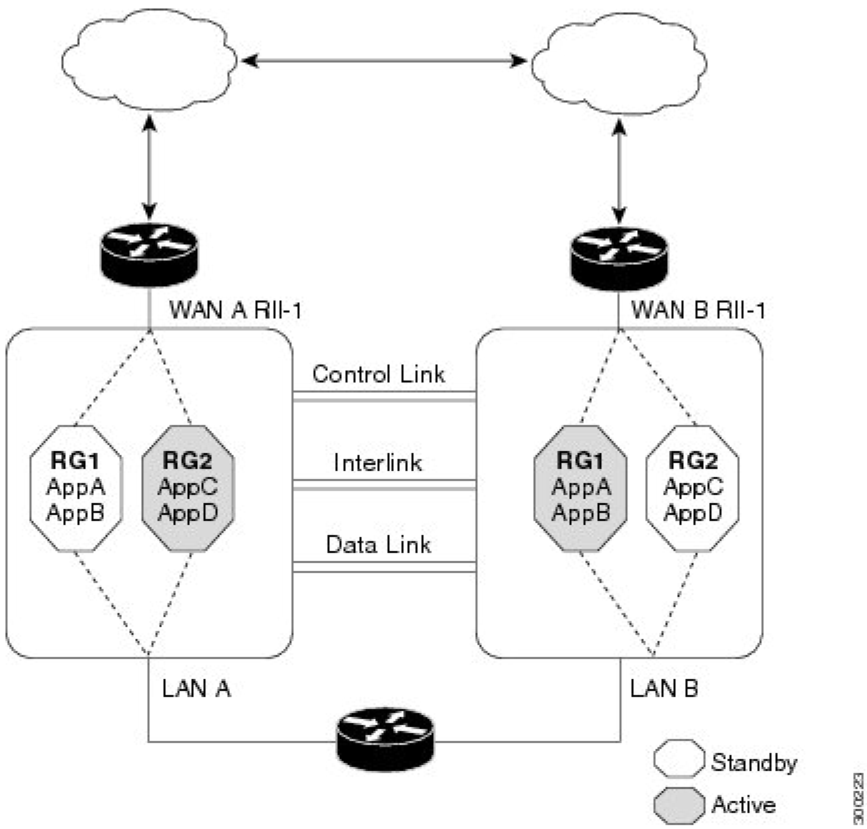
Asymmetric Routing Overview

Asymmetric routing occurs when packets from TCP or UDP connections flow in different directions through different routes. In asymmetric routing, packets that belong to a single TCP or UDP connection are forwarded through one interface in a redundancy group (RG), but returned through another interface in the same RG. In asymmetric routing, the packet flow remains in the same RG. When you configure asymmetric routing, packets received on the standby RG are redirected to the active RG for processing. If asymmetric routing is not configured, the packets received on the standby RG may be dropped.

Asymmetric routing determines the RG for a particular traffic flow. The state of the RG is critical in determining the handling of packets. If an RG is active, normal packet processing is performed. In case the RG is in a standby state and you have configured asymmetric routing and the **asymmetric-routing always-divert enable** command, packets are diverted to the active RG. Use the **asymmetric-routing always-divert enable** command to always divert packets received from the standby RG to the active RG.

The figure below shows an asymmetric routing scenario with a separate asymmetric-routing interlink interface to divert packets to the active RG.

Figure 14: Asymmetric Routing Scenario



The following rules apply to asymmetric routing:

- 1:1 mapping exists between the redundancy interface identifier (RII) and the interface.
- 1:*n* mapping exists between the interface and an RG. (An asymmetric routing interface can receive traffic from and send traffic to multiple RGs. For a non asymmetric-routing interface (normal LAN interface), a 1:1 mapping exists between the interface and the RG.)
- 1:*n* mapping exists between an RG and applications that use it. (Multiple applications can use the same RG).
- 1:1 mapping exists between an RG and the traffic flow. The traffic flow must map only to a single RG. If a traffic flow maps to multiple RGs, an error occurs.
- 1:1 or 1:*n* mapping can exist between an RG and an asymmetric-routing interlink as long as the interlink has sufficient bandwidth to support all the RG interlink traffic.

Asymmetric routing consists of an interlink interface that handles all traffic that is to be diverted. The bandwidth of the asymmetric-routing interlink interface must be large enough to handle all expected traffic that is to be diverted. An IPv4 address must be configured on the asymmetric-routing interlink interface, and the IP address of the asymmetric routing interface must be reachable from this interface.

Note We recommend that the asymmetric-routing interlink interface be used for interlink traffic only and not be shared with high availability control or data interfaces because the amount of traffic on the asymmetric-routing interlink interface could be quite high.

Asymmetric Routing Support in Firewalls

For intrabox asymmetric routing support, the firewall does a stateful Layer 3 and Layer 4 inspection of Internet Control Message Protocol (ICMP), TCP, and UDP packets. The firewall does a stateful inspection of TCP packets by verifying the window size and order of packets. The firewall also requires the state information from both directions of the traffic for stateful inspection. The firewall does a limited inspection of ICMP information flows. It verifies the sequence number associated with the ICMP echo request and response. The firewall does not synchronize any packet flows to the standby redundancy group (RG) until a session is established for that packet. An established session is a three-way handshake for TCP, the second packet for UDP, and informational messages for ICMP. All ICMP flows are sent to the active RG.

The firewall does a stateless verification of policies for packets that do not belong to the ICMP, TCP, and UDP protocols.

The firewall depends on bidirectional traffic to determine when a packet flow should be aged out and diverts all inspected packet flows to the active RG. Packet flows that have a pass policy and that include the same zone with no policy or a drop policy are not diverted.

Note The firewall does not support the **asymmetric-routing always-divert enable** command that diverts packets received on the standby RG to the active RG. By default, the firewall forces all packet flows to be diverted to the active RG.

Active/Active Failover

In an active/active failover configuration, both devices can process network traffic. Active/active failover generates virtual MAC (VMAC) addresses for interfaces in each redundancy group (RG).

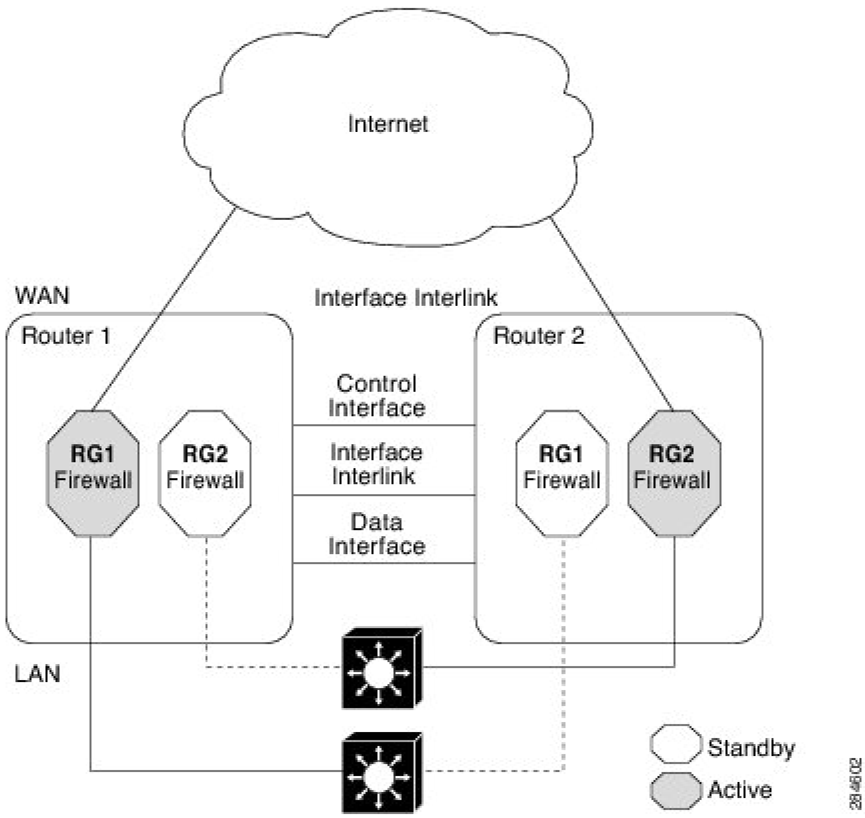
One device in an active/active failover pair is designated as the primary (active) device, and the other is designated as the secondary (standby) device. Unlike with active/standby failover, this designation does not indicate which device becomes active when both devices start simultaneously. Instead, the primary/secondary designation determines the following:

- The device that provides the running configuration to the failover pair when they start simultaneously.
- The device on which the failover RG appears in the active state when devices start simultaneously. Each failover RG in the configuration is configured with a primary or secondary device preference. You can configure both failover RGs to be in the active state on a single device and the standby failover RGs to be on the other device. You can also configure one failover RG to be in the active state and the other RG to be in the standby state on a single device.

Active/Active Load-Sharing Application Redundancy

The following figure shows two RGs, RG1 and RG2. The firewall is registered to both the groups. RG1 has a high priority on Router 1 and RG2 on Router 2. The firewall will process half of the sessions through RG1 on Router 1 and the other half through RG2 on Router 2. As a result, the firewall actively processes traffic on both routers.

Figure 15: Active/Active Load-Sharing Application Redundancy



In an enterprise scenario, if all WAN links on Router 1 fail, switchover happens on Router 2. For example, if there is only one WAN link per box, the failure of the WAN link on the active RG triggers a failover. In the case of a hardware or software failure such as Cisco software reload, the standby will detect active groups on the failed router either through the hello packets timeout or through Bidirectional Forwarding Detection (BFD) if BFD is configured on the control interface.

When Router 1 goes down in the scenarios described, the standby RG will assume the active role on Router 2. When the RG changes the state from standby to active, the firewall will change the state of all sessions in the new active RG and will start processing the traffic.

Active/Standby Failover

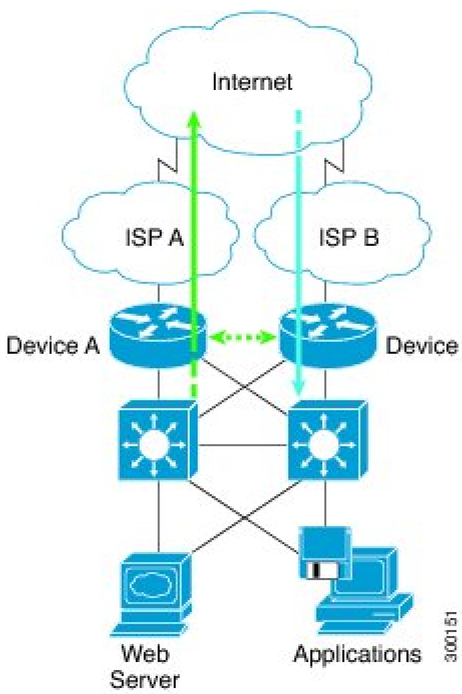
Active/standby failover enables you to use a standby device to take over the functionality of a failed device. A failed active device changes to the standby state, and the standby device changes to the active state. The device that is now in the active state takes over IP addresses and MAC addresses of the failed device and starts processing traffic. The device that is now in the standby state takes over standby IP addresses and MAC addresses. Because network devices do not see any change in the MAC-to-IP address pairing, Address Resolution Protocol (ARP) entries do not change or time out anywhere on the network.

In an active/standby scenario, the main difference between two devices in a failover pair depends on which device is active and which device is a standby, namely which IP addresses to use and which device actively passes the traffic. The active device always becomes the active device if both devices start up at the same time (and are of equal operational health). MAC addresses of the active device are always paired with active IP addresses.

Asymmetric Routing in a WAN-LAN Topology

Asymmetric routing supports only a WAN-LAN topology. In a WAN-LAN topology, devices are connected through LAN interfaces on the inside and WAN interfaces on the outside. There is no control on the routing of return traffic received through WAN links. Asymmetric routing controls the routing of return traffic received through WAN links in a WAN-LAN topology. The figure below shows a WAN-LAN topology.

Figure 16: Asymmetric Routing in a WAN-LAN Topology



Exclusive Virtual IP Addresses and Exclusive Virtual MAC Addresses

Virtual IP (VIP) addresses and virtual MAC (VMAC) addresses are used by security applications to control interfaces that receive traffic. An interface is paired with another interface, and these interfaces are associated with the same redundancy group (RG). The interface that is associated with an active RG exclusively owns the VIP and VMAC. The Address Resolution Protocol (ARP) process on the active device sends ARP replies for any ARP request for the VIP, and the Ethernet controller for the interface is programmed to receive packets destined for the VMAC. When an RG failover occurs, the ownership of the VIP and VMAC changes. The interface that is associated with the newly active RG sends a gratuitous ARP and programs the interface's Ethernet controller to accept packets destined for the VMAC.

IPv6 Support

You can assign each redundancy group (RG) on a traffic interface for both IPv4 and IPv6 virtual IP (VIP) addresses under the same redundancy interface identifier (RII). Each RG uses a unique virtual MAC (VMAC) address per RII. For an RG, the IPv6 link-local VIP and global VIP coexist on an interface.

You can configure an IPv4 VIP, a link-local IPv6 VIP, and/or a global IPv6 VIP for each RG on a traffic interface. IPv6 link-local VIP is mainly used when configuring static or default routes, whereas IPv6 global VIP is widely used in both LAN and WAN topologies.

You must configure a physical IP address before configuring an IPv4 VIP.

Checkpoint Facility Support for Application Redundancy

Checkpointing is the process of storing the current state of a device and using that information during restart when the device fails. The checkpoint facility (CF) supports communication between peers by using the Inter-Process Communication (IPC) protocol and the IP-based Stream Control Transmission Protocol (SCTP). CF also provides an infrastructure for clients or devices to communicate with their peers in multiple domains. Devices can send checkpoint messages from the active to the standby device.

Application redundancy supports multiple domains (also called groups) that can reside within the same chassis and across chassis. Devices that are registered to multiple groups can send checkpoint messages from one group to their peer group. Application redundancy supports interchassis domain communication. Checkpointing happens from an active group to a standby group. Any combination of groups can exist across chassis. The communication across chassis is through SCTP transport over a data link interface that is dedicated to application redundancy.

Note Domains in the same chassis cannot communicate with each other.

How to Configure Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls

Configuring a Firewall

In this task, you will do the following:

- Configure a firewall.
- Create a security source zone.
- Create a security destination zone.
- Create a security zone pair by using the configured source and destination zones.
- Configure an interface as a zone member.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** {**icmp** | **tcp** | **udp**}
5. **exit**
6. **parameter-map type inspect global**
7. **redundancy**
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** *class-map-name*
11. **inspect**
12. **exit**
13. **class class-default**
14. **drop**
15. **exit**
16. **exit**
17. **zone security** *security-zone-name*
18. **exit**
19. **zone security** *security-zone-name*
20. **exit**
21. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
22. **service-policy type inspect** *policy-map-name*
23. **exit**
24. **interface** *type number*
25. **ip address** *ip-address mask*
26. **encapsulation dot1q** *vlan-id*
27. **zone-member security** *security-zone-name*
28. **end**
29. To attach a zone to another interface, repeat Steps 21 to 25.

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any ddos-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** {**icmp** | **tcp** | **udp**} **Example:** `Device(config-cmap)# match protocol tcp`	Configures the match criterion for a class map based on the specified protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 6	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Defines a global inspect parameter map and enters parameter-map type inspect configuration mode.
Step 7	**redundancy** **Example:** `Device(config-profile)# redundancy`	Enables firewall high availability.
Step 8	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 9	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ddos-fw`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 10	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ddos-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.

	Command or Action	Purpose
Step 11	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 13	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Configures the default class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 14	**drop** **Example:** `Device(config-pmap-c)# drop`	Allows traffic to pass between two interfaces in the same zone.
Step 15	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 16	**exit** **Example:** `Device(config-pmap)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 17	**zone security** *security-zone-name* **Example:** `Device(config)# zone security private`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair—a source and a destination zone.
Step 18	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 19	**zone security** *security-zone-name* **Example:** `Device(config)# zone security public`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair—a source and a destination zone.

	Command or Action	Purpose
Step 20	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 21	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security private2public source private destination public`	Creates a zone pair and enters security zone-pair configuration mode.
Step 22	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect ddos-fw`	Attaches a policy map to a top-level policy map.
Step 23	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 24	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/0.1`	Configures an interface and enters subinterface configuration mode.
Step 25	**ip address** *ip-address mask* **Example:** `Device(config-subif)# ip address 10.1.1.1 255.255.255.0`	Configures an IP address for the subinterface.
Step 26	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 2`	Sets the encapsulation method used by the interface.
Step 27	**zone-member security** *security-zone-name* **Example:** `Device(config-subif)# zone-member security private`	Configures the interface as a zone member. • For the *security-zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the device or initiated by the device) is dropped by default. To permit traffic through an interface that is a zone member, you

	Command or Action	Purpose
		must make that zone part of a zone pair to which you apply a policy. If the policy permits traffic (via **inspect** or **pass** actions), traffic can flow through the interface.
Step 28	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.
Step 29	To attach a zone to another interface, repeat Steps 21 to 25.	—

Configuring a Redundancy Application Group and a Redundancy Group Protocol

Redundancy groups consist of the following configuration elements:

- The amount by which the priority will be decremented for each object.
- Faults (objects) that decrement the priority
- Failover priority
- Failover threshold
- Group instance
- Group name
- Initialization delay timer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **priority** *value* [**failover threshold** *value*]
8. **preempt**
9. **track** *object-number* **decrement** *number*
10. **exit**
11. **protocol** *id*
12. **timers hellotime** {*seconds* | **msec** *msec*} **holdtime** {*seconds* | **msec** *msec*}
13. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] | **key-chain** *key-chain-name*}
14. **bfd**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group and enters redundancy application group configuration mode.

	Command or Action	Purpose
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	Specifies an optional alias for the protocol instance.
Step 7	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	Specifies the initial priority and failover threshold for a redundancy group.
Step 8	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the redundancy group and enables the standby device to preempt the active device. • The standby device preempts only when its priority is higher than that of the active device.
Step 9	**track** *object-number* **decrement** *number* **Example:** `Device(config-red-app-grp)# track 50 decrement 50`	Specifies the priority value of a redundancy group that will be decremented if an event occurs on the tracked object.
Step 10	**exit** **Example:** `Device(config-red-app-grp)# exit`	Exits redundancy application group configuration mode and enters redundancy application configuration mode.
Step 11	**protocol** *id* **Example:** `Device(config-red-app)# protocol 1`	Specifies the protocol instance that will be attached to a control interface and enters redundancy application protocol configuration mode.
Step 12	**timers hellotime** {*seconds* | **msec** *msec*} **holdtime** {*seconds* | **msec** *msec*} **Example:** `Device(config-red-app-prtcl)# timers hellotime 3 holdtime 10`	Specifies the interval between hello messages sent and the time period before which a device is declared to be down. • Holdtime should be at least three times the hellotime.
Step 13	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] | **key-chain** *key-chain-name*} **Example:** `Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`	Specifies authentication information.

	Command or Action	Purpose
Step 14	**bfd** **Example:** `Device(config-red-app-prtcl)# bfd`	Enables the integration of the failover protocol running on the control interface with the Bidirectional Forwarding Detection (BFD) protocol to achieve failure detection in milliseconds. • BFD is enabled by default.
Step 15	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring Data, Control, and Asymmetric Routing Interfaces

In this task, you configure the following redundancy group (RG) elements:

- The interface that is used as the control interface.
- The interface that is used as the data interface.
- The interface that is used for asymmetric routing. This is an optional task. Perform this task only if you are configuring asymmetric routing for Network Address Translation (NAT).

Note Asymmetric routing, data, and control must be configured on separate interfaces for zone-based firewall. However, for Network Address Translation (NAT), asymmetric routing, data, and control can be configured on the same interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **data** *interface-type interface-number*
7. **control** *interface-type interface-number* **protocol** *id*
8. **timers delay** *seconds* [**reload** *seconds*]
9. **asymmetric-routing interface** *type number*
10. **asymmetric-routing always-divert enable**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group (RG) and enters redundancy application group configuration mode.
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/1`	Specifies the data interface that is used by the RG.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control GigabitEthernet 1/0/0 protocol 1`	Specifies the control interface that is used by the RG. • The control interface is also associated with an instance of the control interface protocol.
Step 8	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 100 reload 400`	Specifies the time required for an RG to delay role negotiations that start after a fault occurs or the system is reloaded.
Step 9	**asymmetric-routing interface** *type number* **Example:** `Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 0/1/1`	Specifies the asymmetric routing interface that is used by the RG.

	Command or Action	Purpose
Step 10	**asymmetric-routing always-divert enable** **Example:** `Device(config-red-app-grp)# asymmetric-routing always-divert enable`	Always diverts packets received from the standby RG to the active RG.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface

Note

- You must not configure a redundant interface identifier (RII) on an interface that is configured either as a data interface or as a control interface.
- You must configure the RII and asymmetric routing on both active and standby devices.
- You cannot enable asymmetric routing on the interface that has a virtual IP address configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **redundancy rii** *id*
5. **redundancy group** *id* [**decrement** *number*]
6. **redundancy asymmetric-routing enable**
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/3`	Selects an interface to be associated with the redundancy group (RG) and enters interface configuration mode.
Step 4	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 600`	Configures the redundancy interface identifier (RII).
Step 5	**redundancy group** *id* [**decrement** *number*] **Example:** `Device(config-if)# redundancy group 1 decrement 20`	Enables the RG redundancy traffic interface configuration and specifies the amount to be decremented from the priority when the interface goes down. **Note** You need not configure an RG on the traffic interface on which asymmetric routing is enabled.
Step 6	**redundancy asymmetric-routing enable** **Example:** `Device(config-if)# redundancy asymmetric-routing enable`	Establishes an asymmetric flow diversion tunnel for each RG.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls

Example: Configuring a Firewall

```
Router# configure terminal
Router(config)# class-map type inspect match-any ddos-class
Router(config-cmap)# match protocol tcp
Router(config-cmap-c)# exit
Router(config)# parameter-map type inspect global
Router(config-profile)# redundancy
```

```
Router(config-profile)# exit
Router(config)# policy-map type inspect ddos-fw
Router(config-pmap)# class type inspect ddos-class
Router(config-pmap-c)# inspect
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# zone security private
Router(config-sec-zone)# exit
Router(config)# zone security public
Router(config-sec-zone)# exit
Router(config)# zone-pair security private2public source private destination public
Router((config-sec-zone-pair)# service-policy type inspect ddos-fw
Router((config-sec-zone-pair)# exit
Router(config)# interface gigabitethernet 0/1/0.1
Router(config-subif)# ip address 10.1.1.1 255.255.255.0
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# zone-member security private
Router(config-subif)# exit
Router(config)# interface gigabitethernet 1/1/0.1
Router(config-subif)# ip address 10.2.2.2 255.255.255.0
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# zone-member security public
Router(config-subif)# end
```

Example: Configuring a Redundancy Application Group and a Redundancy Group Protocol

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name group1
Device(config-red-app-grp)# priority 100 failover threshold 50
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# track 50 decrement 50
Device(config-red-app-grp)# exit
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 10
Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100
Device(config-red-app-prtcl)# bfd
Device(config-red-app-prtcl)# end
```

Example: Configuring Data, Control, and Asymmetric Routing Interfaces

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/1
Device(config-red-app-grp)# control GigabitEthernet 1/0/0 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 0/1/1
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# end
```

Example: Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/3
Device(config-if)# redundancy rii 600
Device(config-if)# redundancy group 1 decrement 20
Device(config-if)# redundancy asymmetric-routing enable
Device(config-if)# end
```

Additional References

Related Documents

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

Technical Assistance

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

Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls

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

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

Table 5: Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls

Feature Name	Releases	Feature Information
Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls	15.2(3)T	The Interchassis Asymmetric Routing Support for Zone-Based Policy Firewalls feature supports the forwarding of packets from a standby redundancy group to the active redundancy group for packet handling. The following commands were introduced or modified: **asymmetric-routing**, **debug redundancy application group asymmetric-routing**, **redundancy asymmetric-routing enable**, **redundancy group (interface)**, **redundancy rii**, and **show redundancy application asymmetric-routing**.

CHAPTER 6

WAAS Support in Zone-Based Firewalls

Zone-based firewalls support Wide Area Application Services (WAAS). WAAS allows the 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.

This module provides more information about the WAAS Support in Zone-Based Firewalls feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Restrictions for WAAS Support in Zone-Based Firewalls

The following restrictions apply to this feature:

- In a Wide-Area Application Services (WAAS) and firewall configuration, all packets processed by a Wide Area Application Engine (WAE) must pass through the firewall in both directions to support the Web Cache Coordination Protocol (WCCP). This situation occurs because the Layer 2 redirect is not available in Cisco IOS 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.

- In a WAAS and firewall configuration, WCCP does not support traffic redirection using policy-based routing (PBR).

Information About WAAS Support in Zone-Based Firewalls

WAAS Support for the Cisco Firewall

Depending on your release, the Wide Area Application Services (WAAS) firewall software provides an integrated firewall that optimizes security-compliant WANs and application acceleration solutions with the following benefits:

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

WAAS has an automatic discovery mechanism that uses TCP options during the initial three-way handshake 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 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 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.

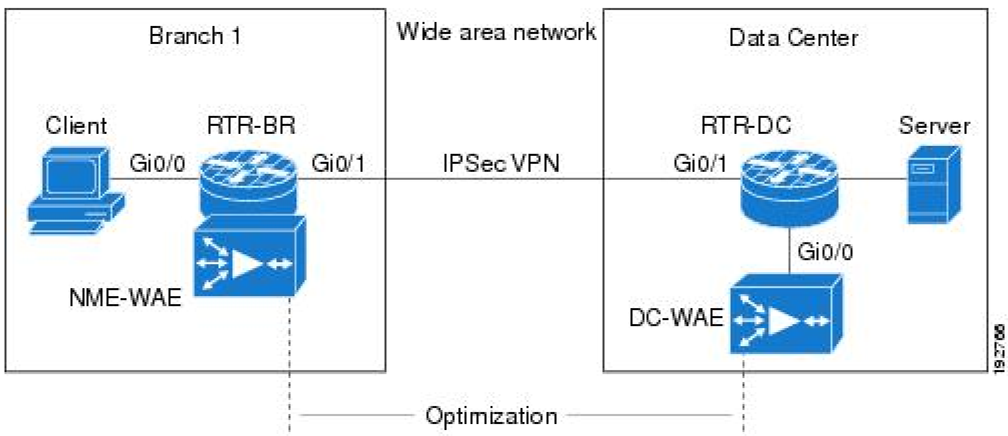
Note Stateful Layer 7 inspection on the client side can also be performed on nonoptimized traffic.

WAAS Traffic Flow Optimization Deployment Scenarios

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

The figure below shows an example of an end-to-end WAAS traffic flow optimization with the Cisco firewall. In this particular deployment, a Network Management Equipment (NME)-WAE device is on the same device as the Cisco firewall. Web Cache Communication Protocol (WCCP) is used to redirect traffic for interception.

Figure 17: End-to-End WAAS Optimization Path

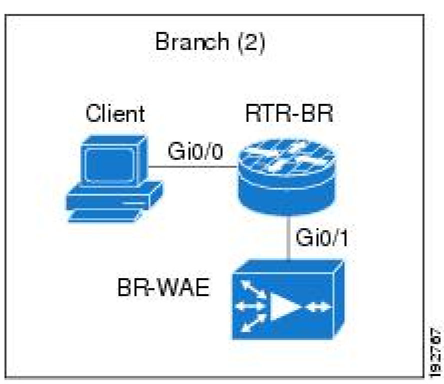


WAAS Branch Deployment with an Off-Path Device

A Wide Area Application Engine (WAE) device can be either a standalone WAE device or an NME-WAE that is installed on an Integrated Services Router (ISR) as an integrated service engine (as shown in the figure Wide Area Application Service [WAAS] Branch Deployment).

The figure below shows a WAAS branch deployment that uses Web Cache Communication Protocol (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.

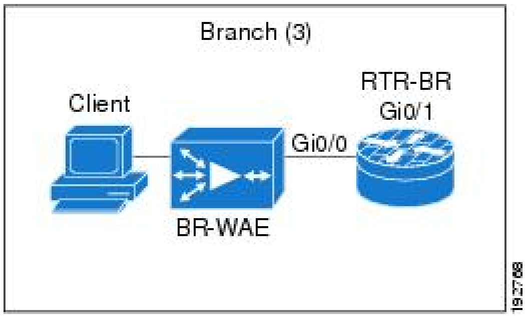
Figure 18: WAAS Off-Path Branch Deployment



WAAS Branch Deployment with an Inline Device

The figure below shows a Wide Area Application Service (WAAS) branch deployment that has an inline Wide Area Application Engine (WAE) device that is physically in front of the Integrated Services Router (ISR). Because the WAE device is in front of the device, the Cisco firewall receives WAAS optimized packets, and as a result, Layer 7 inspection on the client side is not supported.

Figure 19: WAAS Inline Path Branch Deployment



An edge WAAS device with the Cisco firewall is applied at branch office sites that must inspect the traffic moving to and from a WAN connection. The Cisco 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 and maintaining security while accommodating WAAS optimization advantages.

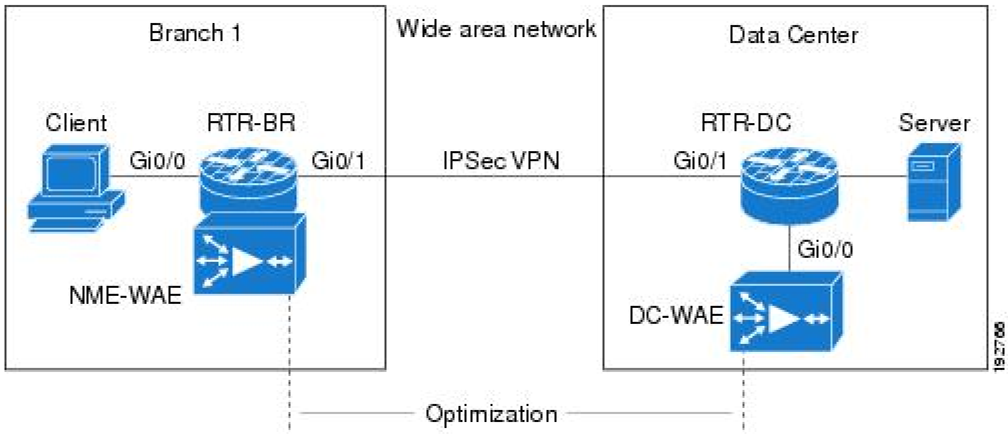
Note If the WAE device is in the inline location, the device enters its bypass mode after the automatic discovery process. Although the device is not directly involved in WAAS optimization, the device must be aware that WAAS optimization is applied to the traffic in order to apply the Cisco firewall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

WAAS and Firewall Integration Support

The following sections describe three different WAAS traffic flow optimization scenarios for branch office deployments. WAAS traffic flow optimization works with the Cisco IOS XE firewall feature on Cisco Aggregation Services Routers (ASRs).

The figure below shows an example of an end-to-end WAAS traffic flow optimization with the Cisco IOS XE firewall. In this particular deployment, an NME-WAE device is on the Cisco IOS Integrated Services Router (ISR).

Figure 20: End-to-End WAAS Optimization Path



WCCP is used to redirect traffic for interception. NME-WAE is not supported on ASR. Therefore, to support NME-WAE in the branch office must be an ISR.

How to Configure WAAS Support in Zone-Based Firewalls

Configuring a Parameter Map for WAAS Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp** *service-id*
4. **ip wccp** *service-id*
5. **parameter-map type inspect global**
6. **waas enable**
7. **log dropped-packets enable**
8. **max-incomplete low**
9. **max-incomplete high**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip wccp** *service-id* **Example:** `Device(config)# ip wccp 61`	Enters the Web Cache Communication Protocol (WCCP) dynamically defined service identifier number.
Step 4	**ip wccp** *service-id* **Example:** `Device(config)# ip wccp 62`	Enters the WCCP dynamically defined service identifier number.
Step 5	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Defines a global inspect parameter map and enters parameter-map type inspect configuration mode.
Step 6	**waas enable** **Example:** `Device(config-profile)# waas enable`	Enables Wide-Area Application Services (WAAS) Express on a WAN interface.
Step 7	**log dropped-packets enable** **Example:** `Device(config-profile)# log dropped-packets enable`	Logs the packets dropped by the firewall.
Step 8	**max-incomplete low** **Example:** `Device(config)# max-incomplete low 18000`	Defines the maximum number of half-open sessions; after which the firewall stops deleting half-open sessions.
Step 9	**max-incomplete high** **Example:** `Device(config)# max-incomplete high 20000`	Defines the maximum number of half-open sessions that can enter a network; after which the firewall starts deleting half-open sessions.

	Command or Action	Purpose
Step 10	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Configuring Class Maps and Policy Maps for WAAS Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-name*
4. **match protocol** *protocol-name* [**signature**]
5. **match protocol** *protocol-name* [**signature**]
6. **match protocol** *protocol-name* [**signature**]
7. **match protocol** *protocol-name* [**signature**]
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class-map type inspect** *class-name*
11. **inspect**
12. **exit**
13. **class class-default**
14. **drop**
15. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**class-map type inspect match-any** *class-name* **Example:** `Device(config)# class-map type inspect match-any most-traffic`	Creates an inspect type class map for the traffic class and enters class-map configuration mode.
Step 4	**match protocol** *protocol-name* [**signature**] **Example:** `Device(config-cmap)# match protocol icmp`	Configures match criteria for a class map on the basis of the specified protocol. • Only Cisco stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 5	**match protocol** *protocol-name* [**signature**] **Example:** `Device(config-cmap)# match protocol ftp`	Configures match criteria for a class map on the basis of a specified protocol.
Step 6	**match protocol** *protocol-name* [**signature**] **Example:** `Device(config-cmap)# match protocol tcp`	Configures match criteria for a class map on the basis of a specified protocol.
Step 7	**match protocol** *protocol-name* [**signature**] **Example:** `Device(config-cmap)# match protocol udp`	Configures match criteria for a class map on the basis of a specified protocol.
Step 8	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
Step 9	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p1`	Creates a Layer 3 and Layer 4 inspect type policy map and enters policy-map configuration mode.
Step 10	**class-map type inspect** *class-name* **Example:** `Device(config-pmap)# class-map type inspect most-traffic`	Specifies the firewall traffic (class) map on which an action is to be performed and enters policy-map class configuration mode.
Step 11	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables Cisco stateful packet inspection.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 13	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies the matching of the system default class. • If the system default class is not specified, unclassified packets are matched.
Step 14	**drop** **Example:** `Device(config-pmap-c)# drop`	Drops packets that are sent to a device.
Step 15	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to global configuration mode.

Configuring Zones and Zone-Pairs for WAAS Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone security** *zone-name*
8. **exit**
9. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
10. **service-policy type inspect** *policy-map-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*
14. **exit**
15. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
16. **service-policy type inspect** *policy-map-name*
17. **exit**
18. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
19. **service-policy type inspect** *p-----*
20. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**zone security** *zone-name* **Example:** `Device(config)# zone security in`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** *zone-name* **Example:** `Device(config)# zone security out`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone security** *zone-name* **Example:** `Device(config)# zone security waas`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 8	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 9	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security in-out source in destination out`	Creates a zone pair and enters security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 10	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p1`	Attaches a firewall policy map to a zone-pair.
Step 11	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 12	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security out-in source out destination in`	Creates a zone pair and enters security zone-pair configuration mode.
Step 13	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p1`	Attaches a firewall policy map to a zone-pair.
Step 14	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 15	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security waas-out source waas destination out`	Creates a zone pair and enters security zone-pair configuration mode.
Step 16	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p1`	Attaches a firewall policy map to a zone-pair.
Step 17	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 18	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security in-waas source in destination waas`	Creates a zone pair and enters security zone-pair configuration mode.
Step 19	**service-policy type inspect** *p-----* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p1`	Attaches a firewall policy map to a zone-pair.

	Command or Action	Purpose
Step 20	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to privileged EXEC mode.

Configuring Interfaces for WAAS Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **description** *line-of-description*
5. **no ip dhcp client request tftp-server-address**
6. **no ip dhcp client request router**
7. **ip address dhcp**
8. **ip wccp** *service-identifier* **redirect in**
9. **ip wccp** *service-identifier* **redirect in**
10. **ip flow ingress**
11. **ip nat outside**
12. **ip virtual-reassembly in**
13. **ip virtual-reassembly out**
14. **zone-member security** *zone-name*
15. **load-interval** *seconds*
16. **delay** *throughput-delay*
17. **duplex auto**
18. **speed auto**
19. **exit**
20. **interface** *type number*
21. **description** *line-of-description*
22. **ip address** *ip-address mask*
23. **ip pim spare-mode**
24. **ip nat inside**
25. **ip virtual-reassembly in**
26. **zone-member security** *zone-name*
27. **ip igmp version**{**1** | **2** | **3**}
28. **delay** *tens-of-microseconds*
29. **duplex auto**
30. **speed auto**
31. **exit**
32. **interface** *type number*
33. **description** *line-of-description*
34. **ip address** *ip-address mask*
35. **ip wccp redirect exclude in**
36. **ip nat inside**
37. **ip virtual-reassembly in**
38. **zone-member security** *zone-name*
39. **load-interval** *seconds*

40. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**description** *line-of-description* **Example:** `Device(config-if)# description WAN connection`	(Optional) Describes an interface.
Step 5	**no ip dhcp client request tftp-server-address** **Example:** `Device(config-if)# no ip dhcp client request tftp-server-address`	Removes an option from the Dynamic Host Control Protocol (DHCP) server.
Step 6	**no ip dhcp client request router** **Example:** `Device(config-if)# no ip dhcp client request router`	Removes the default router option from the DHCP server.
Step 7	**ip address dhcp** **Example:** `Device(config-if)# ip address dhcp`	Acquires an IP address on an interface from DHCP.
Step 8	**ip wccp** *service-identifier* **redirect in** **Example:** `Device(config-if)# ip wccp 62 redirect in`	Redirects inbound packets that have the specified dynamic service identifier to the Web Cache Communication Protocol (WCCP) engine.
Step 9	**ip wccp** *service-identifier* **redirect in** **Example:** `Device(config-if)# ip wccp 61 redirect out`	Redirects outbound packets that have the specified dynamic service identifier to the Web Cache Communication Protocol (WCCP) engine.

	Command or Action	Purpose
Step 10	**ip flow ingress** **Example:** `Device(config-if)# ip flow ingress`	Enables NetFlow accounting for traffic that is received on an interface.
Step 11	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Specifies that an interface is connected to the outside network.
Step 12	**ip virtual-reassembly in** **Example:** `Device(config-if)# ip virtual-reassembly in`	
Step 13	**ip virtual-reassembly out** **Example:** `Device(config-if)# ip virtual-reassembly out`	Enables virtual fragment reassembly (VFR) on outbound interface traffic.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security out`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except the traffic bound for the device or initiated by the device) 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 15	**load-interval** *seconds* **Example:** `Device(config-if)# load-interval 30`	Changes the length of time for which data is used to compute load statistics.
Step 16	**delay** *throughput-delay* **Example:** `Device(config-if)# delay 30`	Sets a throughput delay value for an interface.
Step 17	**duplex auto** **Example:** `Device(config-if)# duplex auto`	Enables autonegotiation on an interface. • The interface automatically operates at half-duplex or full-duplex mode depending on environmental factors, such as the type of media and the transmission speeds for the peer routers, hubs, and switches used in the network configuration.

	Command or Action	Purpose
Step 18	**speed auto** **Example:** `Device(config-if)# speed auto`	Enables autonegotiation on an interface.
Step 19	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 20	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies an interface and enters interface configuration mode.
Step 21	**description** *line-of-description* **Example:** `Device(config-if)# description clients`	(Optional) Describes an interface.
Step 22	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.25.50.1 255.255.255.0`	Specifies an IP address for the interface.
Step 23	**ip pim spare-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enables Protocol Independent Multicast (PIM) sparse mode of operation on an interface.
Step 24	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Specifies that an interface is connected to the inside network.
Step 25	**ip virtual-reassembly in** **Example:** `Device(config-if)# ip virtual-reassembly in`	Enables VFR on inbound interface traffic.
Step 26	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security out`	Assigns an interface to a specified security zone.
Step 27	**ip igmp version**{**1** | **2** | **3**} **Example:** `Device(config-if)# ip igmp version 3`	Configure Version 3 of Internet Group Management Protocol (IGMP) on the router.

	Command or Action	Purpose
Step 28	**delay** *tens-of-microseconds* **Example:** `Device(config-if)# delay 30`	Sets a delay value for an interface.
Step 29	**duplex auto** **Example:** `Device(config-if)# duplex auto`	Enables autonegotiation on an interface. • The interface automatically operates at half-duplex or full-duplex mode depending on environmental factors, such as the type of media and the transmission speeds for the peer routers, hubs, and switches used in the network configuration.
Step 30	**speed auto** **Example:** `Device(config-if)# speed auto`	Enables autonegotiation on an interface.
Step 31	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 32	**interface** *type number* **Example:** `Device(config)# interface vlan 1`	Specifies an interface and enters interface configuration mode.
Step 33	**description** *line-of-description* **Example:** `Device(config-if)# description WAAS interface`	(Optional) Describes an interface.
Step 34	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.25.60.1 255.255.255.0`	Specifies an IP address for an interface.
Step 35	**ip wccp redirect exclude in** **Example:** `Device(config-if)# ip wccp redirect exclude in`	Excludes inbound packets from outbound redirection.
Step 36	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Specifies that an interface is connected to the inside network.

	Command or Action	Purpose
Step 37	**ip virtual-reassembly in** **Example:** `Device(config-if)# ip virtual-reassembly in`	Enables VFR on inbound interface traffic.
Step 38	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security waas`	Assigns an interface to a specified security zone.
Step 39	**load-interval** *seconds* **Example:** `Device(config-if)# load-interval 30`	Changes the length of time for which data is used to compute load statistics.
Step 40	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Configuring WAAS for Zone-Based Firewalls

Note Perform this task on the Wide Area Application Engine (WAE) and not on the router on which zone-based firewall is configured.

SUMMARY STEPS

1. **enable**
2. **configure**
3. **primary-interface** *type number*
4. **interface** *type number*
5. **ip address** *ip-address ip-subnet*
6. **exit**
7. **ip default-gateway** *ip-address*
8. **wccp router-list** *number ip-address*
9. **wccp tcp-promiscuousservice-pair** *serviceID serviceID+1*
10. **router-list-num** *number*
11. **redirect-method** {**gre** | **L2**}
12. **egress-method** {**ip-forwarding** | **generic-gre** | **L2** | **wccp-gre**}
13. **enable**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure** **Example:** `Device# configure`	Enters global configuration mode.
Step 3	**primary-interface** *type number* **Example:** `Device(config)# primary-interface Virtual 1/0`	Configures the primary interface for a Wide Area Application Services (WAAS) device.
Step 4	**interface** *type number* **Example:** `Device(config)# interface Virtual 1/0`	Configures an interface and enters interface configuration mode.
Step 5	**ip address** *ip-address ip-subnet* **Example:** `Device(config-if)# ip address 172.25.60.12 255.255.255.0`	Configures the IP address for the interface.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**ip default-gateway** *ip-address* **Example:** `Device(config)# ip default-gateway 172.25.60.1`	Specifies the default gateway.
Step 8	**wccp router-list** *number ip-address* **Example:** `Device(config)# wccp router-list 1 172.25.60.1`	Configures the IP address and router list number for Web Cache Control Protocol (WCCP) Version 2.
Step 9	**wccp tcp-promiscuousservice-pair** *serviceID serviceID+1* **Example:** `Device(config)# wccp tcp-promiscuous service-pair 61 62`	Configures the Web Cache Coordination Protocol (WCCP) Version 2 TCP promiscuous mode service and enters WCCP configuration mode.
Step 10	**router-list-num** *number* **Example:** `Device(config-wccp-service)# router-list-num 1`	Associates a configured router list with the WCCP service on a WAE.
Step 11	**redirect-method** {**gre** | **L2**} **Example:** `Device(config-wccp-service)# redirect-method gre`	Configures the WAE to use Layer 3 GRE packet redirection.
Step 12	**egress-method** {**ip-forwarding** | **generic-gre** | **L2** | **wccp-gre**} **Example:** `Device(config-wccp-service)# egress-method ip-forwarding`	Configures the IP forwarding egress method.
Step 13	**enable** **Example:** `Device(config-wccp-service)# enable`	Enables the WCCP service.
Step 14	**end** **Example:** `Device(config-wccp-service)# end`	Exits WCCP configuration mode and returns to privileged EXEC mode.

Configuration Examples for WAAS Support in Zone-Based Firewalls

Example: Configuring the Cisco Firewall with WAAS

The following is a sample of an end-to-end Wide Area Application Services (WAAS) traffic flow optimization configuration for the firewall that uses Web Cache Communication Protocol (WCCP) to redirect traffic to a Wide Area Application Engine (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.

```
! Zone-based firewall configuration on your router.
ip wccp 61
ip wccp 62
parameter-map type inspect global
 WAAS enable
 log dropped-packets enable
 max-incomplete low 18000
 max-incomplete high 20000
!
class-map type inspect match-any 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
  drop
!
zone security in
!
zone security out
!
zone security waas
!
zone-pair security in-out source in destination out
 service-policy type inspect p1
!
zone-pair security out-in source out destination in
 service-policy type inspect p1
!
zone-pair security waas-out source waas destination out
 service-policy type inspect p1
!
zone-pair security in-waas source in destination waas
 service-policy type inspect p1
!
interface GigabitEthernet0/0
 description WAN Connection
 no ip dhcp client request tftp-server-address
 no ip dhcp client request router
 ip address dhcp
 ip wccp 62 redirect in
 ip wccp 61 redirect out
 ip flow ingress
 ip nat outside
 ip virtual-reassembly in
```

```
 ip virtual-reassembly out
 zone-member security out
 load-interval 30
 delay 30
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 description Clients
 ip address 172.25.50.1 255.255.255.0
 ip pim sparse-mode
 ip nat inside
 ip virtual-reassembly in
 zone-member security in
 ip igmp version 3
 delay 30
 duplex auto
 speed auto
!
interface Vlan1
 description WAAS Interface
 ip address 172.25.60.1 255.255.255.0
 ip wccp redirect exclude in
 ip nat inside
 ip virtual-reassembly in
 zone-member security waas
 load-interval 30
!
```

The following example shows the configuration on the WAE for zone-based firewall support:

Note This configuration cannot be done on the router; but only on the WAE.

```
!Configuration on the WAE.
primary-interface Virtual 1/0
interface Virtual 1/0
 ip address 172.25.60.12 255.255.255.0
!
ip default-gateway 172.25.60.1
wccp router-list 1 172.25.60.1
wccp tcp-promiscuous service-pair 61 62
 router-list-num 1
 redirect-method gre
 egress-method ip-forwarding
 enable
!
```

Note The new configuration, depending on your release, places an 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-Engine 1/0
 ip address 10.70.100.1 255.255.255.252
 ip wccp redirect exclude in
 zone-member security z-waas
```

Additional References for WAAS Support in Zone-Based Firewalls

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
WAAS commands	http://www.cisco.com/c/en/us/support/routers/wide-area-application-services-waas-software/products-command-reference-list.html

Technical Assistance

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

Feature Information for WAAS Support in Zone-Based Firewalls

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

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

Table 6: Feature Information for WAAS Support in Zone-Based Firewalls

Feature Name	Releases	Feature Information
WAAS Support in Zone-Based Firewalls	12.4(15)T	Zone-based firewalls support Wide Area Application Services (WAAS) 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.

CHAPTER 7

Zone-Based Firewall Logging Export Using NetFlow

Zone-based firewalls support the logging of messages to an external collector using NetFlow Version 9 export format. NetFlow Version 9 export format uses templates to define the format of data that is exported. Template records are sent to the collector along with data records, and the collector interprets these records by using the structural information available in the template.

This module describes the various firewall logging counters and how to configure NetFlow Version 9 flow exporter for firewall message logging.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Restrictions for Zone-Based Firewall Logging Export Using NetFlow

The following features are not supported:

- NetFlow-based logging of pass events
- Layer 7 inspection events
- IPFIX and NetFlow Version 5
- Export of records to multiple collectors
- IPv6 events

Information About Zone-Based Firewall Logging Export Using NetFlow

NetFlow Version 9 Logging Overview

Log messages help the monitoring or managing system to report, analyze, and correlate various events for network administrators. With the introduction of the Zone-Based Firewall Logging Export Using NetFlow feature, firewalls also support the export of record templates and events in Cisco NetFlow Version 9 export format.

Zone-based firewalls export some events (audits and alerts) to an external collector using NetFlow Version 9 export format. NetFlow is a Cisco proprietary network protocol that collects IP traffic to gather flow information, events, and statistics on a device and exports this information to a collector device as NetFlow records. The basic output of NetFlow is a flow record. The latest NetFlow flow-record format is NetFlow Version 9. NetFlow Version 9 format uses templates to define the format of the data that is exported. As template records are sent to an external collector along with data records, the collector can interpret the data records using the structural information available in templates.

NetFlow Version 9 records provide the following features:

- Provides templates to format logging events that help collectors to consume and interpret data based on templates.
- Data is binary-coded and easy to encode and decode (parse).
- Scales better than traditional syslogs and provides better logging performance on the device and the management station.

For more information about NetFlow Version 9, see *RFC 3954*.

Note An external collector application is required to parse templates and interpret the logged data for reporting and display.

Firewall Logging Events

Zone-based firewalls export the following event types by using NetFlow Version 9 export format:

- Audit Events—Start Audit Record and Stop Audit Record. Logs messages when sessions are created and deleted.
- Drop Events—Packet Drop notifications. Logs messages when the following events are dropped—unknown protocols, unseen flows, Out-of-Order (OoO) packets, and so on.
- Alert Events—TCP Half Open Alert, Half Open Session Alert, Maximum-Open sessions. Logs TCP half-open alert messages when the TCP half-open alert threshold values exceed the configured limit.

NetFlow Version 9 Start Audit Records

This template describes the format of data records associated with Start Audit events. Records are generated when a firewall creates a new IPv4-to-IPv4 session. A record is created for every new flow that the firewall creates. The Start Record event is similar to the firewall syslog message (SESS_AUDIT_TRAIL_START).

Table 7: NetFlow Version 9 Start Audit Records

Field IDs	Type	Length	Description
FW_SRC_ADDR_IPV4	8	4	Source IPv4 address.
FW_DST_ADDR_IPV4	12	4	Destination IPv4 address.
FW_SRC_PORT	7	2	Source port.
FW_DST_PORT	11	2	Destination port.
FW_PROTOCOL	4	1	Internet Protocol value. Valid values are the following: • 0—Unknown • 0x01—Layer 4 Internet Control Message Protocol (ICMP) • 0x06—Layer 4 TCP • 0x11—Layer 4 UDP
FW_ICMP_TYPE	176	1	ICMP type value that is set only for ICMP packets (for all other packets the value is zero).
FW_ICMP_CODE	177	1	ICMP code value. **Note** This field is not supported by Cisco IOS zone-based firewalls. The value of this field is zero.

Field IDs	Type	Length	Description
FW_EVENT	233	1	Indicates a firewall event. Valid values are the following: • 0—Ignore (invalid) • 1—Flow created • 2—Flow deleted • 3—Flow denied • 4—Flow alert • 5—Flow update
FW_IPV4_IDENT	54	4	IPv4 ID. The value of the ID field in IPv4 packet. If no fragment header is available, the value is zero.
FW_TCP_SEQ	184	4	TCP sequence number.
FW_TCP_ACK	185	4	TCP acknowledgment sequence number. This value is zero for session creation.
FW_TCP_FLAGS	6	1	TCP flags.
FW_EVENT_TIME_MSEC	323	8	Time, in milliseconds, (time since 0000 hours Consolidated Universal Time [UTC] January 1, 1970) when the event occurred.
FW_INITIATOR_OCTETS	231	8	Size of the Layer 4 payload (in bytes) sent by the initiator.
FW_RESPONDER_OCTETS	232	8	Size of the Layer 4 payload (in bytes) arrived from the responder. This value is zero for session creation.
FW_EXT_EVENT	35001	2	Firewall feature extended event code. The values are defined in Table 8.
FW_L7_PROTOCOL_ID	95	4	Layer 7 protocol ID. This field is specified as per RFC 6758. This field consists of two parts: • 1-byte of Classification Engine ID • 3-bytes of Selector ID
FW_XLATE_SRC_ADDR_IPV4	225	4	Translated source IPv4 address.
FW_XLATE_DST_ADDR_IPV4	226	4	Translated destination IPv4 address.

Field IDs	Type	Length	Description
FW_XLATE_SRC_PORT	227	2	Translated source port.
FW_XLATE_DST_PORT	228	2	Translated destination port.
FW_SRC_INTF_ID	10	2	Source interface ifIndex.
FW_DST_INTF_ID	14	2	Destination interface ifIndex.
FW_SRC_VRF_ID	234	4	Ingress virtual routing and forwarding (VRF) ID. This value is zero if there is no VRF configuration on the source interface.
FW_DST_VRF_ID	235	4	Egress VRF ID. This value is zero if there is no VRF configuration on the destination interface.
FLOW_CLASS -or- FW_CLASS_ID	51	4	Class map ID (numeric representation of the class-map name) associated with this flow.
FW_ZONEPAIR_ID	35007	4	Zone pair ID (numeric representation of zone-pair name) associated with this flow.
FW_CTS_SRC_SGT	34000	2	Source security group tag (SGT) (if a match on SGT) for this flow.

NetFlow Version 9 Stop Audit Records

This template describes the format of data records associated with the Stop Audit event. This record is generated when a firewall deletes an existing IPv4-to-IPv4 session. This record is generated for every flow that is deleted or terminated by a firewall. This event is similar to the firewall syslog message (SESS_AUDIT_TRAIL).

Note The export of this event is not rate limited.

Table 8: NetFlow Version 9 Stop Audit Records

Field IDs	Type	Length	Description
FW_SRC_ADDR_IPV4	8	4	Source IPv4 address.
FW_DST_ADDR_IPV4	12	4	Destination IPv4 address.
FW_SRC_PORT	7	2	Source port.

Field IDs	Type	Length	Description
FW_DST_PORT	11	2	Destination port.
FW_PROTOCOL	4	1	Internet Protocol value. Valid values are the following: • 0—Unknown • 0x01—Layer 4 Internet Control Message Protocol (ICMP) • 0x06—Layer 4 TCP • 0x11—Layer 4 UDP
FW_ICMP_TYPE	176	1	ICMP type value. The value is set only for ICMP packets; the value of all other packets is zero.
FW_ICMP_CODE	177	1	ICMP code value. **Note** This field is not supported by Cisco IOS zone-based firewalls. The value of this field is zero.
FW_EVENT	233	1	Indicates a firewall event. Valid values are the following: • 0—Ignore (invalid) • 1—Flow created • 2—Flow deleted • 3—Flow denied • 4—Flow alert • 5—Flow update
FW_IPV4_IDENT	54	4	IPv4 identification. This value is zero for a Stop Audit event.
FW_TCP_SEQ	184	4	TCP sequence number.
FW_TCP_ACK	185	4	TCP acknowledgment sequence number.
FW_TCP_FLAGS	6	1	TCP flags.
FW_EVENT_TIME_MSEC	323	8	Time, in milliseconds, (time since 0000 hours Consolidated Universal Time [UTC] January 1, 1970) when the event occurred.

Field IDs	Type	Length	Description
FW_INITIATOR_OCTETS	231	8	Size of the Layer 4 payload (in bytes) sent by the initiator.
FW_RESPONDER_OCTETS	232	8	Size of the Layer 4 payload (in bytes) arrived from the responder.
FW_EXT_EVENT	35001	2	Firewall feature extended event code. The values are defined in Table 8.
FW_L7_PROTOCOL_ID	95	4	Layer 7 protocol ID as specified in RFC 6758. This ID consists of two parts: • 1-byte of Classification Engine ID • 3-bytes of Selector ID
FW_XLATE_SRC_ADDR_IPV4	225	4	Translated source IPv4 address.
FW_XLATE_DST_ADDR_IPV4	226	4	Translated destination IPv4 address.
FW_XLATE_SRC_PORT	227	2	Translated source port.
FW_XLATE_DST_PORT	228	2	Translated destination port.
FW_SRC_INTF_ID	10	2	Source interface ifIndex.
FW_DST_INTF_ID	14	2	Destination interface ifIndex.
FW_SRC_VRF_ID	234	4	Ingress virtual routing and forwarding (VRF) ID. This value is zero if there is no VRF configuration on the source interface.
FW_DST_VRF_ID	235	4	Egress VRF ID. This value is zero if there is no VRF configuration on the destination interface.
FLOW_CLASS or FW_CLASS_ID	51	4	Class map ID associated with this flow.
FW_ZONEPAIR_ID	35007	4	Zone pair ID associated with this flow.
FW_CTS_SRC_SGT	34000	2	Source security group tag (SGT) (if a match on SGT) for this flow.

NetFlow Version 9 Flow-Denied Records

This template describes the format of the data records associated with a flow-denied event. This record is generated when a firewall denies an IPv4-to-IPv4 flow or packet. This record is generated for every flow that is denied or packet that is dropped by the firewall. The FW_EXT_EVENT specifies the reason for the flow drop or denial. This event matches the syslog message DROP_PKT.

Table 9: NetFlow Version 9 Flow-Denied Records

Field IDs	Type	Length	Description
FW_SRC_ADDR_IPV4	8	4	Source IPv4 address.
FW_DST_ADDR_IPV4	12	4	Destination IPv4 address.
FW_SRC_PORT	7	2	Source port.
FW_DST_PORT	11	2	Destination port.
FW_PROTOCOL	4	1	Internet Protocol value. Valid values are the following: • 0—Unknown • 0x01—Layer 4 Internet Control Message Protocol (ICMP) • 0x06—Layer 4 TCP • 0x11—Layer 4 UDP
FW_ICMP_TYPE	176	1	ICMP type value that is set only for ICMP packets (for all other packets the value is zero).
FW_ICMP_CODE	177	1	ICMP code value. **Note** This field is not supported by Cisco IOS zone-based firewalls. The value of this field is zero.

Field IDs	Type	Length	Description
FW_EVENT	233	1	Indicates a firewall event. Valid values are the following: • 0—Ignore (invalid) • 1—Flow created • 2—Flow deleted • 3—Flow denied • 4—Flow alert • 5—Flow update
FW_IPV4_IDENT	54	4	IPv4 ID. The value of the ID field in an IPv4 packet. If no fragment header is available, the value is zero.
FW_TCP_SEQ	184	4	TCP sequence number.
FW_TCP_ACK	185	4	TCP acknowledgment sequence number. This value is zero for session creation.
FW_TCP_FLAGS	6	1	TCP flags.
FW_EVENT_TIME_MSEC	323	8	Time, in milliseconds, (time since 0000 hours Consolidated Universal Time [UTC] January 1, 1970) when the event occurred.
FW_INITIATOR_OCTETS	231	8	Size of the Layer 4 payload (in bytes) sent by the initiator.
FW_RESPONDER_OCTETS	232	8	Size of the Layer 4 payload (in bytes) arrived from the responder. This value is zero for session creation.
FW_EXT_EVENT	35001	2	Firewall feature extended event code. The values are defined in Table 8.

Field IDs	Type	Length	Description
FW_L7_PROTOCOL_ID	95	4	Layer 7 protocol ID. This field is specified as per RFC 6758. This field consists of two parts: • 1-byte of Classification Engine ID • 3-bytes of Selector ID
FW_XLATE_SRC_ADDR_IPV4	225	4	Translated source IPv4 address.
FW_XLATE_DST_ADDR_IPV4	226	4	Translated destination IPv4 address.
FW_XLATE_SRC_PORT	227	2	Translated source port.
FW_XLATE_DST_PORT	228	2	Translated destination port.
FW_SRC_INTF_ID	10	2	Source interface ifIndex.
FW_DST_INTF_ID	14	2	Destination interface ifIndex.
FW_SRC_VRF_ID	234	4	Ingress virtual routing and forwarding (VRF) ID. This value is zero if there is no VRF configuration on the source interface.
FW_DST_VRF_ID	235	4	Egress VRF ID. This value is zero if there is no VRF configuration on the destination interface.
FLOW_CLASS or FW_CLASS_ID	51	4	Class map ID (numeric representation of the class-map name) associated with this flow.
FW_ZONEPAIR_ID	35007	4	Zone pair ID (numeric representation of zone-pair name) associated with this flow.
FW_CTS_SRC_SGT	34000	2	Source security group tag (SGT) (if a match on SGT) for this flow.

TCP Half-Open Alert Records

Zone-based firewalls provide protection for hosts against denial-of-service (DoS) attacks such as TCP SYN-flood attack. The threshold values to detect this event can be set using the following commands:

```
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp max-incomplete host 100
```

or

```
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp max-incomplete host 100 block-time 10
```

When the threshold values exceed the configured limit, the information for this event is exported as TCP Half-Open Alert Record. A TCP session that has not yet reached the established state is called a half-open session. The two scenarios that trigger the export of this record are the following:

- TCP maximum-incomplete value is configured, and block time is not configured. When the maximum number of half-open sessions that reach a host exceeds the configured limit, the firewall generates NetFlow logs with the FW_EXT_EVENT set to FW_EXT_ALERT_HOST_TCP_ALERT_ON. This event is similar to firewall syslog message ID HOST_TCP_ALERT_ON.
- TCP maximum-incomplete value and block time are configured:
    - When the maximum number of half-open sessions that reach a host exceeds the configured limit, the firewall blocks all subsequent TCP connection requests. After the configured blocking interval expires, TCP connection requests are allowed. NetFlow logs FW_EXT_EVENT that is set to FW_EXT_ALERT_BLOCK_HOST and FW_BLACKOUT_SECS (indicates the blocking interval in seconds). This event is similar to the syslog message ID BLOCK_HOST.
    - When the blocking interval expires and the firewall allows further connections to the host, NetFlow logs FW_EXT_EVENT that is set to FW_EXT_ALERT_UNBLOCK_HOST and FW_BLACKOUT_SECS. This event is similar to the syslog message ID UNBLOCK_HOST.

Note The export of this event is not rate limited.

Table 10: TCP Half-Open Alert Records

Field ID	Type	Length	Offset	Description
FW_DST_ADDR_IPV4	12	4	0 to 3	Destination IPv4 address.
FW_PROTOCOL	4	1	4	Internet Protocol value or ID.
FW_EVENT	233	1	5	High level event code. A value is 4 indicates a flow alert.

Field ID	Type	Length	Offset	Description
FW_EXT_EVENT	35001	2	6 to 7	Extended firewall event code. Valid values are the following: • 0x1E—FW_EXT_ALERT_UNBLOCK_HOST • 0x1F—FW_EXT_ALERT_HOST_TCP_ALERT_ON • 0x20—FW_EXT_ALERT_BLOCK_HOST
FW_EVENT_TIME_MSEC	323	8	8 to 15	Time, in milliseconds, (time since 0000 hours Consolidated Universal Time [UTC] January 1, 1970) when the event occurred.
FW_HALFOPEN_CNT	35012	4	16 to 19	Number of half-open TCP sessions.
FW_BLACKOUT_SECS	35004	4	20 to 23	Time duration, in seconds, when a destination is blacked out or unavailable.
FW_DST_INTF_ID	14	2	24 to 26	SNMP ifIndex of the egress interface.
FW_DST_VRF_ID	235	4	27 to 30	Unique ID of the destination virtual routing and forwarding (VRF) instance.
FLOW_CLASS or FW_CLASS_ID	51	4	31 to 34	Class map ID associated with this flow.
FW_ZONEPAIR_ID	35007	4	35 to 38	Zone pair ID associated with this flow.

Half-Open Session Alert Records

This template describes the format of data records for Half Open Session Alert. This record is generated when the number of existing half-open sessions exceed the configured high limit value or drop below the low bound value. The export of this event is not rate limited.

Use the following commands to configure the half-open session limit:

```
Device(config)# parameter-map type inspect param-name
Device(config-profile)# max-incomplete high 20000
Device(config-profile)# max-incomplete low 10000
```

Table 11: Half-Open Session Alert Records

Field ID	Type	Length	Description
FW_EVENT	233	1	High level event code. A value of 4 indicates Flow Alert.
FW_EXT_EVENT	35001	2	Extended Firewall event code. Valid values are the following: • 0x21—FW_EXT_SESS_RATE_ALERT_ON • 0x22—FW_EXT_SESS_RATE_ALERT_OFF
FW_EVENT_TIME_MSEC	323	8	Time, in milliseconds, (time since 0000 hours Consolidated Universal Time [UTC] January 1, 1970) when the event occurred.
FW_EVENT_LEVEL	33003	1	Extended firewall event code. Valid values are the following: • 0x01—Per box • 0x02—Virtual routing and forwarding (VRF) • 0x03—Zone • 0x04—Class map • Other values are undefined
FW_EVENT_LEVEL_ID	33004	4	Defines the identifier for the FW_EVENT_LEVEL event. Valid values are the following: • 0x02—VRF_ID. • 0x03—ZONE_ID. • 0x04—CLASS_ID. • In all other cases and if FW_EVENT_LEVEL is not present the field ID is zero.
FW_CONFIGURED_VALUE	33005	4	Specifies the configured half-open session high-limit value or low-bound value.

Maximum Session Alert Records

This template describes the format of data records for the Maximum Session Alert event. This record is generated when the number of firewall sessions exceed the configured limit. The export of this event is not rate limited and is generated when sessions exceed the configured limit. Use the following commands to configure the maximum limit for firewall sessions:

```
Device (config)# parameter-map type inspect param-map
Device(config-profile)# sessions maximum 20000
```

Table 12: Maximum Session Alert Records

Field ID	Type	Length	Offset	Description
FW_EVENT	233	1	0	High level event code. A value of 4 indicates flow alert.
FW_EXT_EVENT	35001	2	1 to 2	Extended firewall event code. A value of 0x23 indicates FW_EXT_L4_SESSION_LIMIT.
FW_EVENT_TIME_MSEC	323	8	3 to 10	Time, in milliseconds, (time since 0000 hours Consolidated Universal Time [UTC] 4 January 1, 1970) when the event occurred.
FW_MAX_SESSIONS	35008	4	11 to 14	Maximum sessions allowed for this zone pair or class ID.
FW_ZONEPAIR_ID	35007	4	15 to 18	Zone pair ID associated with this flow.
FLOW_CLASS or FW_CLASS_ID	51	4	19 to 22	Class map ID associated with this flow.

NetFlow Version 9 Option Template Records

This template provides information about the data that is exported as part of data records. For example, a data record exports the Interface-ID field, which is a numerical representation of the interface. To obtain the corresponding name on the device, the device exports option template data records that consists of the Interface-ID-to-Interface-Name value mapping. Option template data records are exported periodically based on the configured option template timeout value.

Protocol ID-to-Name Mapping

The protocol ID-to-name mapping is obtained by exporting the inspect-protocol-table option template and enabling the **debug policy-firewall exporter** command.

The following is sample output from the **debug policy-firewall exporter** command. In the following output, protocol ID is 6xxyyzz where xxyyzz is the 3-byte protocol ID in hexadecimal notation.

```
FW-EXPORT: Sent Opt Rec Protocol Id:(6000001) <--> Name:(ftp)
FW-EXPORT: Sent Opt Rec Protocol Id:(6000002) <--> Name:(telnet)
FW-EXPORT: Sent Opt Rec Protocol Id:(6000003) <--> Name:(smtp)
FW-EXPORT: Sent Opt Rec Protocol Id:(6000004) <--> Name:(http)
FW-EXPORT: Sent Opt Rec Protocol Id:(6000005) <--> Name:(tacacs)
FW-EXPORT: Sent Opt Rec Protocol Id:(6000006) <--> Name:(dns)
FW-EXPORT: Sent Opt Rec Protocol Id:(6000007) <--> Name:(sql-net)
FW-EXPORT: Sent Opt Rec Protocol Id:(6000008) <--> Name:(https)
FW-EXPORT: Sent Opt Rec Protocol Id:(6000009) <--> Name:(tftp)
FW-EXPORT: Sent Opt Rec Protocol Id:(600000A) <--> Name:(gopher)
FW-EXPORT: Sent Opt Rec Protocol Id:(600000B) <--> Name:(finger)
FW-EXPORT: Sent Opt Rec Protocol Id:(600000C) <--> Name:(kerberos)
FW-EXPORT: Sent Opt Rec Protocol Id:(600000D) <--> Name:(pop2)
FW-EXPORT: Sent Opt Rec Protocol Id:(600000E) <--> Name:(pop3)
!
!
!
```

VRF Name Options Record

NetFlow Version 9 supports the export of the vrf-table option template. The external collectors must correlate the virtual routing and forwarding (VRF) IDs in the firewall records with the VRF names specified in vrf-table option records received from the exporter.

The following is a sample output from the **show flow exporter templates** command:

```
Device# show flow exporter templates

Flow Exporter tfoo
  Client: Option options vrf-id-name-table
  Exporter Format: NetFlow Version 9
  Template ID    : 256
  Source ID      : 0
  Record Size    : 40
  Template layout
 _____________________________________________________________________
|                 Field                    |  Type | Offset |  Size  |
--------------------------------------------------------------------
| v9-scope system                          |     1 |     0  |     4  |
| routing vrf input                        |   234 |     4  |     4  |
| routing vrf name                         |   236 |     8  |    32  |
--------------------------------------------------------------------
```

Interface ID-to-Name Mapping

There is no option template to export interface ID-to-name mapping. External collectors must query the ifIndex MIB via Simple Network Management Protocol (SNMP) to correlate SRC_IF_INDEX and DST_IF_INDEX to the interface description or name.

Class-Name Option Records

This template describes the format of option templates that map FW_CLASS_ID to a class name.

Table 13: Class-Name Options Records

Field ID	Type	Length	Offset	Description
v9-scope-system	1	4	0 to 3	Provides information about the NetFlow process to which the option record refers. Valid values are the following: • 0x0001—System • 0x0002—Interface • 0x0003—Line card • 0x0004—NetFlow cache • 0x0005—Template
FLOW_CLASS or FW_CLASS_ID	51	4	4 to 7	Class map ID on the device.
FW_CLASS_NAME	100	64	8 to 71	Name of the class.

Firewall Extended Event Records

The following table describes the FW_EXT_EVENT_ID fields associated with the logging of drop events. The firewall extended event records map extended-event IDs to names.

Table 14: Firewall Extended Event Records

Value	FW_EXT_EVENT_ID	Description
0	INSP_L4_NO_ERROR	No specific extended event.
1	INSP_L4_INVAL_HLEN	Invalid Layer 4 header length.
2	INSP_L4_C3PL_LOOKUP_FAIL	Policy match failure.
3	INSP_L4_POLICE_RATE_LIMIT	Police rate limiting
4	INSP_L4_SESSION_LIMIT	Session limit exceeded.
5	INSP_L4_ICMP_INVAL_RET	Invalid return packet.
6	INSP_L4_ICMP_INVAL_DEST	Invalid destination address for unreachable or time-exceeded packets.
7	INSP_L4_UDP_DISA_BIDIR	Bidirectional traffic disabled.
8	INSP_L4_SYN_INVAL_FLDATA	Synchronize (SYN) packet with data or with push (PSH) or urgent (URG) flags.

Value	FW_EXT_EVENT_ID	Description
9	INSP_L4_INVALID_CONN_SEG	Segment does not match any TCP connection.
10	INSP_L4_INVALID_SEG	Invalid TCP segment.
11	INSP_L4_INVALID_SEQ	Invalid TCP sequence number.
12	INSP_L4_INVALID_ACK	Invalid TCP acknowledgment (ACK) or no ACK.
13	INSP_L4_INVALID_FLAGS	Invalid TCP flags.
14	INSP_L4_INVALID_CHKSM	Invalid TCP checksum.
15	INSP_L4_SYN_IN_WIN	SYN inside current window. A SYN packet is seen within the window of an already established TCP connection.
16	INSP_L4_RST_IN_WIN	Reset (RST) inside current window. An RST packet is seen within the window of an already established TCP connection.
17	INSP_L4_OOO_SEG	Out-of-Order (OoO) segment.
18	INSP_L4_OOO_INVALID_FLAGS	OoO segment with invalid flag.
19	INSP_L4_RETRANS_SEG	Retransmitted segment.
20	INSP_L4_RETRANS_INVALID_FLAGS	Retransmitted segment with invalid flag.
21	INSP_L4_STRAY_SEQ	Stray TCP segment.
22	INSP_L4_INTERNAL_ERR	Firewall internal error.
23	INSP_L4_INVALID_WINDOW_SCALE	Invalid window scale option.
24	INSP_L4_INVALID_TCP_OPTION	Invalid TCP option.
25	INSP_UNKNOWN_ERR	Unknown error.
26	INSP_L4_C3PL_LOOKUP_FAIL_NO_ZONE_PAIR	Lookup failure because zone pairs are not available between zones.
27	INSP_L4_C3PL_LKP_FAIL_ZONE_TO_NONZONE	Lookup failure because only one interface is the member of a zone and other interface is not a member of any zone.
28	INSP_L4_C3PL_LOOKUP_FAIL_NO_POLICY	Policy not present in the zone pair.
29	INSP_L4_DROP_CONFIGURED	Drop action configured in a policy map.

Value	FW_EXT_EVENT_ID	Description
30	FW_EXT_ALERT_UNBLOCK_HOST	Blocking of TCP attempts to a specified host is removed.
31	FW_EXT_ALERT_HOST_TCP_ALERT_ON	Maximum incomplete host limit of half-open TCP connections exceeded. **Note** Once this message is sent to the host, the traffic from that host can be blocked by sending the FW_EXT_ALERT_BLOCK_HOST message for the time period configured.
32	FW_EXT_ALERT_BLOCK_HOST	Maximum incomplete host threshold of half-open TCP connections exceeded.
33	FW_EXT_SESS_RATE_ALERT_ON	Exceeded either the maximum incomplete high threshold of half-open connections or the new connection initiation rate ID.
34	FW_EXT_SESS_RATE_ALERT_OFF	Either the number of half-open connections or the new connection initiation rate is below the maximum incomplete low threshold.
35	FW_EXT_MAX_SESS_LIMIT	Number of established sessions has crossed the configured threshold.

Firewall Extended Event-Named Option Records

This template describes the format of option templates that map FW_EXT_EVENT to an event name or a description

Field ID	Type	Length	Offset	Description
v9-scope-system	1	4	0 to 3	This field provides information about the NetFlow process to which the options record refers. Valid values are the following: • 0x0001—System • 0x0002—Interface • 0x0003—Line card • 0x0004—NetFlow cache • 0x0005—Template
FW_EXT_EVENT	35001	2	4 to 5	Extended event code.

Field ID	Type	Length	Offset	Description
FW_EXT_EVENT_DESC	35010	64	6 to 69	Description of the extended event.

Extended Event ID-to-Name Mapping

The extended event ID-to-name mapping records are obtained by exporting the inspect-ext-event-table option template and enabling the **debug policy-firewall exporter** command.

The following is sample output from the **debug policy-firewall exporter** command:

```
*Dec 20 05:24:50.917: FW-EXPORT: Sent Optional Record Ext Event id:(0x0) <--> Name:(NO_ERROR)
*Dec 20 05:24:50.917: FW-EXPORT: Sent Optional Record Ext Event id:(0x1) <-->
Name:(INVALID_HEADER_LENGTH)
*Dec 20 05:24:50.917: FW-EXPORT: Sent Optional Record Ext Event id:(0x2) <-->
Name:(POLICY_MATCH_FAILURE)
*Dec 20 05:24:50.917: FW-EXPORT: Sent Optional Record Ext Event id:(0x3) <-->
Name:(POLICE_RATE_LIMITING)
*Dec 20 05:24:50.917: FW-EXPORT: Sent Optional Record Ext Event id:(0x4) <-->
Name:(SESSION_LIMITING)
*Dec 20 05:24:50.917: FW-EXPORT: Sent Optional Record Ext Event id:(0x5) <-->
Name:(INVALID_RETURN_PACKET)
!
!
!
```

Protocol-Name Option Records

This template describes the format of option templates that map the FW_PROTOCOL_ID to the protocol name. As per RFC 6759, the protocol ID or application ID (that is, the IANA Flow Field Type 95) is represented as a 4-byte number with the following two parts:

- 1-byte of Classification Engine ID. For NetFlow logging this value is always equal to 6, which specifies that this value is user defined.
- 3-bytes of Selector ID. This value represents the actual protocol ID or application ID as defined by the device.

Note All values are not exported; only protocols that the zone-based firewall supports are exported.

Table 15: Protocol-Name Option Records

Field IDs	Type	Length	Offset	Description
v9-scope-system	1	4	0 to 3	This field refers to the NetFlow process to which the options record refers. Valid values are the following: • 0x0001—System • 0x0002—Interface • 0x0003—Line card • 0x0004—NetFlow cache • 0x0005—Template
FW_L7_PROTOCOL_ID	95	4	4 to 7	Layer 7 protocol ID as specified in RFC 6758. The ID consists of the following two parts: • 1-byte of Classification Engine ID • 3-bytes of Selector ID
FLOW_FIELD_L7_PROTOCOL_NAME	96	64	8 to 72	Specifies the name of the protocol or application.

Zone-Pair Name Option Records

This template describes the format of option templates that map FW_ZONEPAIR_ID event to a zone-pair name configured on the device.

Table 16: Zone-Pair Name Options Records

Field ID	Type	Length	Offset	Description
v9-scope-system	1	4	0 to 3	This field provides information about the NetFlow process to which the options record refers. Valid values are the following: • 0x0001—System • 0x0002—Interface • 0x0003—Line card • 0x0004—NetFlow cache • 0x0005—Template

Field ID	Type	Length	Offset	Description
FW_ZONEPAIR_ID	35007	4	4 to 7	Zone-pair ID configured on the device.
FW_ZONEPAIR_NAME	35009	64	8 to 71	Name of the zone pair that corresponds to the zone-pair ID.

How to Configure Zone-Based Firewall Logging Export Using NetFlow

Perform the following tasks to configure zone-based firewall logging export using NetFlow:

1. Define a flow exporter and option templates.
2. Attach the flow exporter to a global parameter map.

Defining a Flow Exporter and Option Templates

In this task you define the flow exporter and then the option templates. You must attach the flow exporter to a parameter map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *name*
4. **export-protocol netflow-v9**
5. **destination** {*ipv4-address* | *ipv6-address*} [**vrf** *vrf-name*]
6. **transport udp** *port-number*
7. **option inspect-class-table** [**timeout** *timeout-value*]
8. **option inspect-protocol-table** [**timeout** *timeout-value*]
9. **option inspect-ext-event-table** [**timeout** *timeout-value*]
10. **option zone-pair-table** [**timeout** *timeout-value*]
11. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *name* **Example:** `Device(config)# flow exporter v9-flow`	Creates or modifies a Flexible NetFlow flow exporter and enters flow exporter configuration mode.
Step 4	**export-protocol netflow-v9** **Example:** `Device(config-flow-exporter)# export-protocol netflow-v9`	Configures the export protocol for a Flexible NetFlow flow exporter.
Step 5	**destination** {*ipv4-address* | *ipv6-address*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 10.1.1.1`	Configures an export destination for a Flexible NetFlow flow exporter.
Step 6	**transport udp** *port-number* **Example:** `Device(config-flow-exporter)# transport udp 200`	Specifies UDP as the transport protocol for a flow exporter.
Step 7	**option inspect-class-table** [**timeout** *timeout-value*] **Example:** `Device(config-flow-exporter)# option inspect-class-table timeout 2000`	Configures a policy-firewall class table for a flow exporter.
Step 8	**option inspect-protocol-table** [**timeout** *timeout-value*] **Example:** `Device(config-flow-exporter)# option inspect-protocol-table timeout 3000`	Configures a policy-firewall inspect protocol table for a flow exporter.
Step 9	**option inspect-ext-event-table** [**timeout** *timeout-value*] **Example:** `Device(config-flow-exporter)# option inspect-ext-event-table timeout 1200`	Configures a policy-firewall extended event table for a flow exporter.
Step 10	**option zone-pair-table** [**timeout** *timeout-value*] **Example:** `Device(config-flow-exporter)# option zone-pair-table timeout 2500`	Configures a policy-firewall zone-pair table for a flow exporter.
Step 11	**end** **Example:** `Device(config-flow-exporter)# end`	Exits flow exporter configuration mode and returns to privileged EXEC mode.

Attaching a Flow Exporter to a Global Parameter Map

You must attach the NetFlow flow exporter (v9-flow) that you configured to a global parameter map. You cannot attach a flow exporter to a default or user-defined parameter map.

Note After attaching the flow exporter to a global parameter map, you can configure the **audit-trail** command for a default or user-defined parameter map; log messages will be exported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **exporter** *exporter-name*
5. **alert** {**on** | **off**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Configures an inspect-type global parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 4	**exporter** *exporter-name* **Example:** `Device(config-profile)# exporter v9-flow`	Configures a flow exporter. • The flow exporters that you previously configured are listed as options for this command. In this example, you can see v9-flow as an option.

	Command or Action	Purpose
Step 5	**alert** {**on** | **off**} **Example:** `Device(config-profile)# alert on`	Enables or disables the console display of stateful packet inspection alert messages.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to global configuration mode.

Verifying Zone-Based Firewall Logging Export Using NetFlow

Use the following commands to troubleshoot your configuration:

SUMMARY STEPS

1. **enable**
2. **debug policy-firewall exporter**
3. **show parameter-map type inspect global**
4. **show flow exporter** *exporter-name* [**statistics** | **templates**]
5. **show flow exporter** {**templates** | **statistics** | **export-ids netflow-v9**}
6. **show running-config flow exporter export-ids netflow-v9**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:
```
Device> enable
```

Step 2 **debug policy-firewall exporter**

Enables logging of firewall NetFlow Version 9 messages.

Example:
```
Device# debug policy-firewall exporter
```

Step 3 **show parameter-map type inspect global**

Displays global inspect type parameter map values.

Example:

```
Device# show parameter-map type inspect global

alert on
  sessions maximum 2147483647
  waas disabled
  l2-transparent dhcp-passthrough disabled
  log dropped-packets disabled
  log summary disabled
  max-incomplete low 18000
  max-incomplete high 20000
  one-minute low 2147483647
  one-minute high 2147483647
  tcp reset-PSH disabled
  exporter v9-flow
```

Step 4 **show flow exporter** *exporter-name* [**statistics** | **templates**]

Displays the status and statistics for the Flexible NetFlow user-configured flow exporter.

Example:

```
Device# show flow exporter v9-flow

Flow Exporter v9-flow:
  Description:              User defined
  Export protocol:          NetFlow Version 9
  Transport Configuration:
    Destination IP address: 10.1.1.1
    Source IP address:      10.4.5.2
    Transport Protocol:     UDP
    Destination Port:       9995
    Source Port:            0
    DSCP:                   0x0
    TTL:                    255
    Output Features:        Not Used
```

Step 5 **show flow exporter** {**templates** | **statistics** | **export-ids netflow-v9**}

Displays flow exporter statistics.

Example:

```
Device# show flow exporter statistics

Flow Exporter netflow-v9:
  Packet send statistics (last cleared 00:02:27 ago):
    Successfully sent:         0                      (0 bytes)
    No FIB:                    13                     (16010 bytes)

  Client send statistics:
    Client: Option Start audit v4 (session creation)
      Records added:           0
      Bytes added:             0

    Client: Option Stop audit v4 (session deletion)
      Records added:           0
      Bytes added:             0

    Client: Option Drop audit v4 (Pak drop)
      Records added:           0
      Bytes added:             0

    Client: Option Alert TCP halfopen
      Records added:           0
      Bytes added:             0
```

```
Client: Option Alert halfopen
  Records added:          0
  Bytes added:            0

Client: Option Alert max session
  Records added:          0
  Bytes added:            0

Client: Option Template for FW class-id
  Records added:          2
    - failed to send:     2
  Bytes added:            136
    - failed to send:     136

Client: Option Template for FW protocol-id
  Records added:          172
    - failed to send:     172
  Bytes added:            11696
    - failed to send:     11696

Client: Option Template for FW Extended Event
  Records added:          36
    - failed to send:     36
  Bytes added:            2376
```

Step 6 **show running-config flow exporter export-ids netflow-v9**
Displays flow exporter configuration.

Example:
```
Device# show running-config flow exporter export-ids netflow-v9
```

Configuration Examples for Zone-Based Firewall Logging Export Using NetFlow

Example: Defining a Flow Exporter and Option Templates

```
Device# configure terminal
Device(config)# flow exporter v9-flow
Device(config-flow-exporter)# export-protocol netflow-v9
Device(config-flow-exporter)# destination 10.1.1.1
Device(config-flow-exporter)# transport udp 200
Device(config-flow-exporter)# option inspect-class-table timeout 2000
Device(config-flow-exporter)# option inspect-protocol-table timeout 3000
Device(config-flow-exporter)# option inspect-ext-event-table timeout 1200
Device(config-flow-exporter)# option zone-pair-table timeout 2500
Device(config-flow-exporter)# end
```

Example: Attaching a Flow Exporter to a Global Parameter Map

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# exporter v9-flow
Device(config-profile)# alert on
```

```
Device(config-profile)# end
```

Additional References for Zone-Based Firewall Logging Export Using NetFlow

Related Documents

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

Standards and RFCs

Standard/RFC	Title
RFC 792	Internet Control Message Protocol
RFC 3954	Cisco Systems NetFlow Services Export Version 9
RFC 6758	Tunneling of SMTP Message Transfer Priorities

MIBs

MIB	MIBs Link
ifIndex	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

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

Feature Information for Zone-Based Firewall Logging Export Using NetFlow

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

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

Table 17: Feature Information for Zone-Based Firewall Logging Export Using NetFlow

Feature Name	Releases	Feature Information
Zone-Based Firewall Logging Export Using NetFlow	15.4(2)T	Zone-based firewalls support the logging of messages to an external collector using NetFlow Version 9 export format. NetFlow version 9 export format uses templates to define the format of data that is exported. Template records are sent to collector along with data records, the collector interprets these records by using the structural information available in template. The following commands were introduced or modified by this feature: **debug policy-firewall exporter**, **option (FlexibleNetFlow)**, and **show flow internal**.

CHAPTER 8

Cisco IOS Firewall-SIP Enhancements ALG and AIC

Enhanced Session Initiation Protocol (SIP) inspection in the Cisco IOS firewall provides basic SIP inspect functionality (SIP packet inspection and pinholes opening) as well as protocol conformance and application security. These enhancements give you more control than in previous releases on what policies and security checks to apply to SIP traffic and the capability to filter out unwanted messages or users.

The development of additional SIP functionality in Cisco IOS software provides increased support for Cisco Call Manager (CCM), Cisco Call Manager Express (CCME), and Cisco IP-IP Gateway based voice/video systems. Application Layer Gateway (ALG), and Application Inspection and Control (AIC) SIP enhancements also support RFC 3261 and its extensions.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Cisco IOS Firewall-SIP Enhancements ALG and AIC

The following prerequisites apply to the configuration of Cisco IOS Firewall--SIP Enhancements: ALG and AIC.

Hardware Requirements

- One of the following router platforms:
    - Cisco 861, Cisco 881, or Cisco 881G routers
    - Cisco 1700 routers
    - Cisco 1800 routers
    - Cisco 2600 routers
    - Cisco 2800 routers
    - Cisco 3700 routers
    - Cisco 3800 routers
    - Cisco 7200 routers
    - Cisco 7300 routers

Software Requirements

- Cisco IOS Release 12.4(15)XZ or a later release.

Restrictions for Cisco IOS Firewall-SIP Enhancements ALG and AIC

DNS Name Resolution

Although SIP methods can have Domain Name System (DNS) names instead of raw IP addresses, this feature currently does not support DNS names.

Earlier Releases of Cisco IOS Software

Some Cisco IOS releases earlier than Release 12.4(15)XZ may accept the configuration commands for SIP that are shown in this document; however, those earlier versions will not function properly.

Information About Cisco IOS Firewall-SIP Enhancements ALG and AIC

Firewall and SIP Overviews

This section provides an overview of the Cisco IOS firewall and SIP.

Cisco IOS Firewall

The Cisco IOS firewall extends the concept of static access control lists (ACLs) by introducing dynamic ACL entries that open on the basis of the necessary application ports on a specific application and close these ports at the end of the application session. The Cisco IOS firewall achieves this functionality by inspecting the application data, checking for conformance of the application protocol, extracting the relevant port information to create the dynamic ACL entries, and closing these ports at the end of the session. The Cisco IOS firewall is designed to easily allow a new application inspection whenever support is needed.

Session Initiation Protocol

SIP is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method or function on the server and at least one response.

SIP invitations used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. SIP makes use of elements called proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols.

Firewall for SIP Functionality Description

The Firewall for SIP Support feature allows SIP signaling requests to traverse directly between gateways or through a series of proxies to the destination gateway or phone. After the initial request, if the Record-Route header field is not used, subsequent requests can traverse directly to the destination gateway address as specified in the Contact header field. Thus, the Cisco IOS firewall is aware of all surrounding proxies and gateways and allows the following functionality:

- SIP signaling responses can travel the same path as SIP signaling requests.
- Subsequent signaling requests can travel directly to the endpoint (destination gateway).
- Media endpoints can exchange data between each other.

SIP UDP and TCP Support

RFC 3261 is the current RFC for SIP, which replaces RFC 2543. This feature supports the SIP User Datagram Protocol (UDP) and the TCP format for signaling.

SIP Inspection

This section describes the deployment scenarios supported by the Cisco IOS Firewall--SIP, ALG, and AIC Enhancements feature.

Cisco IOS Firewall Between SIP Phones and CCM

The Cisco IOS firewall is located between CCM or CCME and SIP phones. SIP phones are registered to CCM or CCME through the firewall, and any SIP calls from or to the SIP phones pass through the firewall.

Cisco IOS Firewall Between SIP Gateways

The Cisco IOS firewall is located between two SIP gateways, which can be CCM, CCME, or a SIP proxy. Phones are registered with SIP gateways directly. The firewall sees the SIP session or traffic only when there is a SIP call between phones registered to different SIP gateways. In some scenarios an IP-IP gateway can also be configured on the same device as the firewall. With this scenario all the calls between the SIP gateways are terminated in the IP-IP gateway.

Cisco IOS Firewall with Local CCME and Remote CCME/CCCM

The Cisco IOS firewall is located between two SIP gateways, which can be CCM, CCME, or a SIP proxy. One of the gateways is configured on the same device as the firewall. All the phones registered to this gateway are locally inspected by the firewall. The firewall also inspects SIP sessions between the two gateways when there is a SIP call between them. With this scenario the firewall locally inspects SIP phones on one side and SIP gateways on the other side.

Cisco IOS Firewall with Local CCME

The Cisco IOS firewall and CCME is configured on the same device. All the phones registered to the CCME are locally inspected by the firewall. Any SIP call between any of the phones registered will also be inspected by the Cisco IOS firewall.

How to Configure Cisco IOS Firewall-SIP Enhancements ALG and AIC

Configuring a Policy to Allow RFC 3261 Methods

Perform this task to configure a policy to allow basic RFC 3261 methods and block extension methods.

Note The Cisco IOS Firewall--SIP Enhancements: ALG and AIC feature provides essential support for the new SIP methods such as UPDATE and PRACK, as CCM 5.x and CCME 4.x also use these methods.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** *protocol-name* **match-any** *class-map-name*
4. **match request method** *method-name*
5. **exit**
6. **class-map type inspect** *protocol-name* **match-any** *class-map-name*
7. **match request method** *method-name*
8. **exit**
9. **policy-map type inspect** *protocol-name policy-map-name*
10. **class type inspect** *protocol-name class-map-name*
11. **allow**
12. **exit**
13. **class type inspect** *protocol-name class-map-name*
14. **reset**
15. **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	**class-map type inspect** *protocol-name* **match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect sip match-any sip-class1`	Creates an inspect type class map and enters class-map configuration mode.
Step 4	**match request method** *method-name* **Example:** `Router(config-cmap)# match request method invite`	Matches RFC 3261 methods. Methods include the following: • ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, update.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.
Step 6	**class-map type inspect** *protocol-name* **match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect sip match-any sip-class2`	Creates an inspect type class map and enters class-map configuration mode.
Step 7	**match request method** *method-name* **Example:** `Router(config-cmap)# match request method message`	Matches RFC 3261 methods, which include the following: • ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, update.
Step 8	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.
Step 9	**policy-map type inspect** *protocol-name* *policy-map-name* **Example:** `Router(config)# policy-map type inspect sip sip-policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 10	**class type inspect** *protocol-name* *class-map-name* **Example:** `Router(config-pmap)# class type inspect sip sip_class1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 11	**allow** **Example:** `Router(config-pmap-c)# allow`	Allows SIP inspection.
Step 12	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.

	Command or Action	Purpose
Step 13	**class type inspect** *protocol-name* *class-map-name* **Example:** `Router(config-pmap)# class type inspect sip sip-class2`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 14	**reset** **Example:** `Router(config-pmap-c)# reset`	Resets the class map.
Step 15	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.

Configuring a Policy to Block Messages

Perform this task to configure a policy to block SIP messages coming from a particular proxy device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *url-pattern*
5. **exit**
6. **class-map type inspect** *protocol-name class-map-name*
7. **match request header** *field* **regex** *regex-param-map*
8. **exit**
9. **policy-map type inspect** *protocol-name policy-map-name*
10. **class type inspect** *protocol-name class-map-name*
11. **reset**
12. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex unsecure-proxy`	Configures a parameter-map type to match a specific traffic pattern and enters profile configuration mode.
Step 4	**pattern** *url-pattern* **Example:** `Router(config-profile)# pattern “compromised.server.com”`	Matches a call based on the SIP uniform resource identifier (URI).
Step 5	**exit** **Example:** `Router(config-profile)# exit`	Exits profile configuration mode.
Step 6	**class-map type inspect** *protocol-name class-map-name* **Example:** `Router(config)# class-map type inspect sip sip-class`	Creates an inspect type class map and enters class-map configuration mode.
Step 7	**match request header** *field* **regex** *regex-param-map* **Example:** `Router(config-cmap)# match request header Via regex unsecure-proxy`	Configures a class-map type to match a specific request header pattern.
Step 8	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.

	Command or Action	Purpose
Step 9	**policy-map type inspect** *protocol-name policy-map-name* **Example:** `Router(config)# policy-map type inspect sip sip-policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 10	**class type inspect** *protocol-name class-map-name* **Example:** `Router(config-pmap)# class type inspect sip sip-class`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 11	**reset** **Example:** `Router(config-pmap-c)# reset`	Resets the class map.
Step 12	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.

Configuring a 403 Response Alarm

Perform this task to configure a policy to generate an alarm whenever a 403 response is returned.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *url-pattern*
5. **exit**
6. **class-map type inspect** *protocol-name class-map-name*
7. **match response status regex** *regex-parameter-map*
8. **exit**
9. **policy-map type inspect** *protocol-name policy-map-name*
10. **class type inspect** *protocol-name class-map-name*
11. **log**
12. **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 regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex allowed-im-users`	Configures a parameter-map type to match a specific traffic pattern and enters profile configuration mode.
Step 4	**pattern** *url-pattern* **Example:** `Router(config-profile)# pattern “403”`	Matches a call based on the SIP URI.
Step 5	**exit** **Example:** `Router(config-profile)# exit`	Exits profile configuration mode.
Step 6	**class-map type inspect** *protocol-name class-map-name* **Example:** `Router(config)# class-map type inspect sip sip-class`	Creates an inspect type class map and enters class-map configuration mode.
Step 7	**match response status regex** *regex-parameter-map* **Example:** `Router(config-cmap)# match response status regex allowed-im-users`	Configures a class-map type to match a specific response pattern.
Step 8	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.

	Command or Action	Purpose
Step 9	**policy-map type inspect** *protocol-name policy-map-name* **Example:** `Router(config)# policy-map type inspect sip sip-policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 10	**class type inspect** *protocol-name class-map-name* **Example:** `Router(config-pmap)# class type inspect sip sip-class`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 11	**log** **Example:** `Router(config-pmap-c)# log`	Generates a log of messages.
Step 12	**exit** **Example:** `Router(config)# exit`	Exits policy-map class configuration mode.

Limiting Application Messages

Perform this task to configure a policy to rate-limit INVITE messages.

Note While configuring the **rate-limit** command, do not configure the **allow** or **reset** commands. An error message is displayed if you try to configure the **allow** or **reset** commands while configuring the **rate-limit** command and vice versa.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** *protocol-name* **match-any** *class-map-name*
4. **match request method** *method-name*
5. **exit**
6. **policy-map type inspect** *protocol-name policy-map-name*
7. **class type inspect** *protocol-name class-map-name*
8. **rate-limit** *limit-number*
9. **exit**
10. **exit**
11. **class-map type inspect match-any** *class-map-name*
12. **match protocol** *protocol-name*
13. **exit**
14. **policy-map type inspect** *policy-map-name*
15. **class type inspect** *class-map-name*
16. **inspect**
17. **service-policy** *protocol-name policy-map-name*
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	**class-map type inspect** *protocol-name* **match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect sip match-any class-2`	Creates an inspect type class map and enters class-map configuration mode.

	Command or Action	Purpose
Step 4	**match request method** *method-name* **Example:** `Router(config-cmap)# match request method invite`	Matches RFC 3261 methods. Methods include the following: • ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, update.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.
Step 6	**policy-map type inspect** *protocol-name policy-map-name* **Example:** `Router(config)# policy-map type inspect sip policy-2`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *protocol-name class-map-name* **Example:** `Router(config-pmap)# class type inspect sip class-2`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 8	**rate-limit** *limit-number* **Example:** `Router(config-pmap-c)# rate-limit 16`	Limits the number of SIP messages that strike the Cisco IOS firewall every second.
Step 9	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 10	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode and enters global configuration mode.
Step 11	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any class-1`	Creates an inspect type class map and enters class-map configuration mode.

	Command or Action	Purpose
Step 12	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol sip`	Configures the match criterion for a class map on the basis of the specified protocol.
Step 13	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.
Step 14	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect policy-1`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 15	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect class-1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 16	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 17	**service-policy** *protocol-name policy-map-name* **Example:** `Router(config-pmap-c)# service-policy sip policy_2`	Attaches the policy map to the service policy for the interface or virtual circuit.
Step 18	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.

Limiting Application Messages for a Particular Proxy

Perform this task to configure a policy to rate-limit INVITE messages coming for a particular proxy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *url-pattern*
5. **exit**
6. **class-map type inspect** *protocol-name* **match-any** *class-map-name*
7. **match request method** *method-name*
8. **match request header** *field* **regex** *regex-param-map*
9. **exit**
10. **policy-map type inspect** *protocol-name policy-map-name*
11. **class type inspect** *protocol-name class-map-name*
12. **rate-limit** *limit-number*
13. **exit**
14. **exit**
15. **class-map type inspect match-any** *class-map-name*
16. **match protocol** *protocol-name*
17. **exit**
18. **policy-map type inspect** *policy-map-name*
19. **class type inspect** *class-map-name*
20. **inspect**
21. **service-policy** *protocol-name policy-map-name*
22. **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.

	Command or Action	Purpose
Step 3	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex rate-limited-proxy`	Configures a parameter-map type to match a specific traffic pattern and enters profile configuration mode.
Step 4	**pattern** *url-pattern* **Example:** `Router(config-profile)# pattern "compromised.server.com"`	Matches a call based on the SIP URI.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits profile configuration mode.
Step 6	**class-map type inspect** *protocol-name* **match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect sip match-any class_2`	Creates an inspect type class map and enters class-map configuration mode.
Step 7	**match request method** *method-name* **Example:** `Router(config-cmap)# match request method invite`	Matches RFC 3261 methods. Methods include the following: • ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, update.
Step 8	**match request header** *field* **regex** *regex-param-map* **Example:** `Router(config-cmap)# match request header Via regex rate-limited-proxy`	Configures a class-map type to match a specific request header pattern.
Step 9	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.

	Command or Action	Purpose
Step 10	**policy-map type inspect** *protocol-name policy-map-name* **Example:** `Router(config)# policy-map type inspect sip policy-2`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 11	**class type inspect** *protocol-name class-map-name* **Example:** `Router(config-pmap)# class type inspect sip class-2`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 12	**rate-limit** *limit-number* **Example:** `Router(config-pmap-c)# rate-limit 16`	Limits the number of SIP messages that strike the Cisco IOS firewall every second.
Step 13	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 14	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode and enters global configuration mode.
Step 15	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any class-1`	Creates an inspect type class map and enters class-map configuration mode.
Step 16	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol sip`	Configures the match criterion for a class map on the basis of the specified protocol.
Step 17	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.

	Command or Action	Purpose
Step 18	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect policy-1`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 19	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect class-1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 20	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 21	**service-policy** *protocol-name* *policy-map-name* **Example:** `Router(config-pmap-c)# service-policy sip policy-2`	Attaches the policy map to the service policy for the interface or virtual circuit.
Step 22	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.

Verifying and Troubleshooting Cisco IOS Firewall-SIP Enhancements ALG and AIC

The following commands can be used to troubleshoot the Cisco IOS Firewall--SIP Enhancements: ALG and AIC feature:

1. **clear zone-pair**
2. **debug cce**
3. **debug ip inspect**
4. **debug policy-map type inspect**
5. **show policy-map type inspect zone-pair**
6. **show zone-pair security**

Note

Effective with Cisco IOS Release 12.4(20)T, the **debug ip inspect** command is replaced by the **debug policy-firewall** command. See the *Cisco IOS Debug Command Reference* for more information.

Examples

The following is sample output of the **show policy-map type inspect zone-pair** command when the **session** keyword is used.

```
Router# show policy-map type inspect zone-pair session
policy exists on zp zp_test_out_self
 Zone-pair: zp_test_out_self
  Service-policy inspect : test
   Class-map: c_sip (match-any)
   ...
    Number of Established Sessions = 2
    Established Sessions
     Session 6717A7A0 (192.168.105.118:62265)=>(192.168.105.2:5060) sip:udp SIS_OPEN
      Created 00:10:27, Last heard 00:00:03
      Bytes sent (initiator:responder) [35579:14964]
     Session 67179EA0 (192.168.105.119:62266)=>(192.168.105.2:5060) sip:udp SIS_OPEN
      Created 00:10:27, Last heard 00:03:17
      Bytes sent (initiator:responder) [10689:4093]
    Number of Pre-generated Sessions = 7
     Pre-generated Sessions
      Pre-gen session 6717A560 192.168.105.2[1024:65535]=>192.168.105.118[62265:62265]
   sip:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 67179C60 192.168.105.2[1024:65535]=>192.168.105.119[62266:62266]
   sip:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 67176F60 192.168.105.118[1024:65535]=>192.168.105.2[5060:5060]
 sip:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 67176AE0 192.168.105.118[1024:65535]=>192.168.105.2[18318:18318]
   sip-RTP-data:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 671768A0 192.168.105.2[1024:65535]=>192.168.105.118[62495:62495]
   sip-RTP-data:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 671783A0 192.168.105.118[1024:65535]=>192.168.105.2[18319:18319]
   sip-RTCP-data:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 67176420 192.168.105.2[1024:65535]=>192.168.105.118[62496:62496]
   sip-RTCP-data:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
```

The following is sample output of the **show zone-pair security** command.

```
Router# show zone-pair security
Zone-pair name zp_in_out
 Source-Zone inside Destination-Zone outside
 service-policy test
Zone-pair name zp_in_self
 Source-Zone inside Destination-Zone self
 service-policy test
Zone-pair name zp_self_out
```

```
Source-Zone self Destination-Zone outside
service-policy test
```

Configuration Examples for Cisco IOS Firewall-SIP Enhancements ALG and AIC

Example Firewall and SIP Configuration

The following example shows how to configure the Cisco IOS Firewall--SIP Enhancements: ALG and AIC feature when the Cisco IOS firewall is located between two SIP gateways (CCM or CCME), as described in the Cisco IOS Firewall Between SIP Gateways. Some phones are registered to the CCME inside the firewall (inside zone). Other phones are registered to another CCME / CCM outside the firewall (outside zone). Cisco IOS firewall is configured for SIP inspection when there is no IP-IP gateway configured on the firewall device.

```
class-map type inspect sip match-any sip-aic-class
match request method invite
policy-map type inspect sip sip-aic-policy
class type inspect sip sip-aic-class
rate-limit 15
!
policy-map type inspect sip-policy
class type inspect sip-traffic-class
service-policy sip sip-aic-policy
!
class-map type inspect match-any sip-traffic-class
match protocol sip
!
policy-map type inspect sip-policy
class type inspect sip-traffic-class
inspect my-parameters
!
zone security inside
zone security outside
!
interface fastethernet 0
zone-member security inside
interface fastethernet 1
zone-member security outside
!
zone-pair security in-out source inside destination outside
service-policy type inspect sip-policy
!
zone-pair security in-self source inside destination self
service-policy type inspect sip-policy
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS firewall commands	*Cisco IOS Security Command Reference*

Related Topic	Document Title
SIP information and configuration tasks	Configuring Session Initiation Protocol for Voice over IP" module in the *Cisco IOS Voice, Video, and Fax Configuration Guide*
Additional SIP Information	Guide to Cisco Systems VoIP Infrastructure Solution for SIP

MIBs

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

RFCs

RFC	Title
RFC 3261	*SIP: Session Initiation Protocol*

Technical Assistance

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

Feature Information for Cisco IOS Firewall-SIP Enhancements ALG and AIC

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

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

Table 18: Feature Information for Cisco IOS Firewall-SIP Enhancements: ALG and AIC

Feature Name	Releases	Feature Information
Cisco IOS Firewall--SIP Enhancements: ALG and AIC	12.4(15)XZ 12.4(20)T	This feature provides voice security enhancements within the firewall feature set in Cisco IOS software for Release 12.4(15)XZ and later releases. In Release 12.4(15)XZ, this feature was introduced on the Cisco 861, Cisco 881, and Cisco 881G routers. In Release 12.4(20)T, this feature was implemented on the Cisco 1700, Cisco 1800, Cisco 2600, Cisco 2800, Cisco 3700, Cisco 3800, Cisco 7200, and Cisco 7300 routers. The following commands were introduced or modified: **class-map type inspect**, **match protocol**, **match protocol-violation**, **match req-resp**, **match request**, **match response**, **policy-map type inspect**, **rate-limit (firewall)**.

CHAPTER 9

Firewall-H.323 V3 V4 Support

The Firewall H.323 V3 V4 Support feature provides the firewall with support for the H.323 Voice over IP (VoIP) Version 3 and Version 4 protocols. With Version 3 and Version 4 support, features like call signaling (H.225) over User Datagram Protocol (UDP), multiple call signaling over a single TCP connection, T.38 Fax over TCP, and address resolution using border elements are supported. Support for a rate-limiting mechanism to monitor call attempt rate and call aggregation is also introduced and can be enabled.

H.323 is a multiprotocol and multichannel suite. Channel negotiation parameters are embedded inside encoded H.323 control messages. The Base H.323 Application Layer Gateway (ALG) Support feature provides support in firewall environments to process the H.323 control messages.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Cisco IOS Firewall-H.323 V3 V4 Support

Restrictions for Firewall-H.323 V3 V4 Support

General

- Inspection of H.323 signaling over secure (encrypted) channel is not supported.

Cisco ASR 1000 Series Aggregation Services Routers

- Support is provided for gateway terminals using the H.323v4 with H.225v4 and H.245v7 protocols only.
- Backward compatibility is provided for H.323v2 messages only. H.323v1 messages are ignored.
- Multipoint conferencing, managed by the Multipoint Control Unit (MCU), is not supported.
- The T.120 protocol is not supported.
- The firewall support is limited to H.323 Direct Call Signaling and H.225 RAS Call Signaling.

Information About Firewall-H.323 V3 V4 Support

H.323 and H.225 RAS Implementation

H.225 Registration, Admission, and Status (RAS) signaling in Cisco IOS firewalls is a signaling protocol that is used between endpoints (such as gateways) and gatekeepers. The H.225 standard is used by H.323 for call setup. H.225 includes RAS control, which is used to communicate with the gatekeeper. A RAS signaling channel enables connections between the gatekeeper and H.323 endpoints.

H.323 and H.245 Protocol

During the call setup between H.323 terminals, the following protocols are used:

- H.225 Call Signaling
- H.245 Call Control

Both protocol messages contain embedded IP addresses and ports. Any message passing through a router running Cisco IOS firewall must be decoded, inspected, and encoded back to the packet.

In order for an H.323 call to take place, an H.225 connection on TCP port 1720 needs to be opened. When the H.225 connection is opened, the H.245 session is initiated and established. This connection can take place on a separate channel from the H.225 or it can be done using H.245 tunneling on the same H.225 channel whereby the H.245 messages are embedded in the H.225 messages and set on the previously established H.225 channel.

If the H.245 tunneled message is not understood the Cisco IOS firewall cannot translate the message, which causes a failure in media traffic. H.245 FastConnect procedures will not help because FastConnect is terminated as soon as an H.245 tunneled message is sent.

H.323 Version 3 and Version 4 Features Supported

The table below lists the H.323 Version 3 and Version 4 features supported by Cisco IOS firewall. For information on the H.323 standard, see the Standards section.

Note On the ASR 1000 series routers Cisco IOS firewall support is limited to H.323 Direct Call Signaling and H.225 RAS Call Signaling only.

Table 19: H.323 Standards Features Supported by Cisco IOS Firewall

Standard	Features Supported by Cisco IOS Firewall
H.323 Version 3	• Caller ID • Annex E--Protocol for Multiplexed Call Signaling Transport • Annex G--Communication Between Administrative Domains • Generic information transport • Maintaining and reusing connections using call signaling channel • Supplementary services (call hold, call park and call pickup, message waiting indication, and call waiting)

Standard	Features Supported by Cisco IOS Firewall
H.323 Version 4	• Additive registrations • Alternate gatekeepers • Endpoint capacity • Bandwidth management • Usage information reporting • Generic extensibility framework • Indicating desired protocols • Call status reporting • Enhancements to Annex D (Real-Time Fax) • QoS support for H.323 enhancements • Dual Tone Multifrequency (DTMF) digit transmission using Real-Time Protocol (RTP)

Base H.323 ALG Support

The Base H.323 ALG Support feature provides support for ALGs to perform protocol specific issues such as processing embedded IP address and port numbers and extracting connection and session information from control channels and sessions.

Encoded channel-negotiation parameters are embedded in H.323 control messages. In Cisco IOS firewall environments, the system must intercept these messages and invoke the H.323 ALG to process the messages.

The H.323 ALG performs the following tasks to process the messages:

- Intercepts the H.323 control messages on the H.225.0 TCP port 1720 and on the dynamically negotiated H.245 TCP port.
- Decodes the intercepted control messages.
- Parses the decoded control messages, identifies the embedded IP address and port-number pairs and builds action info tokens based on the IP address and port-number pairs.
- Sends the action info tokens to the Cisco IOS firewall for processing.

The Cisco IOS firewall performs the actions indicated by the action info tokens. The actions performed include session and door entry lookup, creation, and deletion, or address and port translation. When the Cisco IOS firewall completes the action, it fills the action-result field in the action-info token, with the translated IP address and port number, or with an action failure indicator. Cisco IOS firewall then adds a flag to indicate if the packet should be dropped or forwarded. Finally, it returns the action info token to the H.323 ALG.

- Receives the modified action info token from the Cisco IOS firewall and either drops or forwards the packet based on information in the action info token.

The table below lists the H.323 control messages processed by the Base H.323 ALG Support feature. For more information on the H.323 standard, see the Standards section.

Table 20: H.323 Control Messages Processed by Base H.323 ALG Support

Protocol	Messages
H.225.0 Call Signalling	• Setup • Alert • Call proceed • Connect • Facility • Progress • Empty • ReleaseComplete • SetupAcknowlege
H.245 Media Control **Note** If tunnelling mode is enabled H.245 messages may be embedded within H.225.0 messages	• OpenLogicalChannel • OpenLogicalAck • CloseLogicalChannel • CloseLogicalAck

Support of Rate Limiting Mechanism

In addition to supporting Version 3 and Version 4 of the H.323 protocol, support is introduced for a rate-limiting mechanism to monitor call attempt rate and call aggregation. Rate limiting is more important for voice applications where gateways and gatekeepers are set up in less secure arrangements such as a Demilitarized Zone (DMZ). A DMZ can be vulnerable to attack from the Internet.

Rate Limiting of H.323 Traffic Messages

Rate limiting of H.323 traffic control messages is based on actions on H.323 class maps. The messages that are to be rate limited are specified through match message statements within the class map. The rate-limit threshold value is specified by a rate limit command, as an action on the H.323 class map. The rate limit command limits the message attempt rate; it limits the number of H.323 messages being sent per second to and from an end point. Rate Limiting can be used to control call attempt rate.

Note While configuring the **rate-limit** command, do not configure the **allow** or **reset** commands. An error message is displayed if you try to configure the **allow** or **reset** commands while configuring the **rate-limit** command and vice versa.

How to Configure Firewall-H.323 V3 V4 Support

Configuring a Firewall Policy for H.323 Traffic

Configuring a Class Map for H.323 Traffic

Perform this task to define the class map that for H.323 traffic that is to be permitted between zones.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** [**match-any** | **match-all**] **class-map-name**
4. **match protocol protocol-name [parameter-map] [signature]**
5. **match protocol h225ras**
6. match protocol h323-annexe
7. match protocol h323-nxg
8. **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.

	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-any h323-traffic-class`	Creates a Layer 3 and Layer 4 (Top Level) inspect type class map and enters class-map configuration mode.
Step 4	**match protocol protocol-name [parameter-map] [signature]** **Example:** `Router(config-cmap)# match protocol h323`	Configures the match criterion for a class map on the basis of the specified protocol.
Step 5	**match protocol h225ras** **Example:** `Router(config-cmap)# match protocol h225ras`	Configures the match criterion for a class map on the basis of a specified protocol. **Note** You should specify the **h225ras** keyword to create a class map for H.225 RAS protocol classification. For a list of supported protocols, use the command-line interface (CLI) help option (**?**) on your platform.
Step 6	match protocol h323-annexe **Example:** `Router(config-cmap)# match protocol h323-annexe`	Enables the inspection of H.323 Protocol Annex E traffic.
Step 7	match protocol h323-nxg **Example:** `Router(config-cmap)# match protocol h323-nxg`	Enables the inspection of H.323 Protocol Annex G traffic.
Step 8	**end** **Example:** `Router(config-cmap)# end`	Exits class-map configuration mode and enters privileged EXEC mode.

Configuring a Policy Map for H.323 Traffic

Perform this task to create a policy map for H.323 traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect policy-map-name**
4. **class type inspect** *class-map-name*
5. **inspect [parameter-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	**policy-map type inspect policy-map-name** **Example:** `Router(config)# policy-map type inspect h323-policy`	Creates a Layer 3 or Layer inspect type policy map.
Step 4	**class type inspect** *class-map-name* **Example:** `Router(config)# class type inspect h323-traffic-class`	Specifies the traffic (class) on which an action is to be performed. **Note** The *class-map-name* value must match the appropriate class map name specified via the **class-map type inspect**command.
Step 5	**inspect [parameter-map-name]** **Example:** `Router(config)# inspect`	Enables Cisco IOS stateful packet inspection. **Note** The actions **drop** or **allow** may also be used instead of the **inspect** command here.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Configuring a Zone-Pair for H.323 Traffic and Applying an H.323 Policy Map

Perform this task to configure a zone-pair for H.323 traffic and to apply an H.323 policy map to the traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-pair-name*
4. **exit**
5. **zone security** *zone-pair-name*
6. **exit**
7. **zone security** *zone-pair-name*
8. **exit**
9. **zone-pair security zone-pair-name** {**source source-zone-name**| **self**} **destination** [**self** | **destination-zone-name**]
10. **service-policy type inspect** *policy-map-name*
11. **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	**zone security** *zone-pair-name* **Example:** `Router(config) zone security in-out`	Specifies the name of the zone-pair and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone) exit`	Exits security zone configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 5	**zone security** *zone-pair-name* **Example:** `Router(config) zone security inside`	Creates the source zone from which traffic originates and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone) exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone security** *zone-pair-name* **Example:** `Router(config) zone security outside`	Creates the destination zone to which the traffic is bound and enters security zone configuration mode.
Step 8	**exit** **Example:** `Router(config-sec-zone) exit`	Enters global configuration mode.
Step 9	**zone-pair security zone-pair-name** {**source source-zone-name**| **self**} **destination** [**self** | **destination-zone-name**] **Example:** `Router(config)# zone-pair security in-out source inside destination outside`	Associates a zone-pair and declares the names of the routers from which traffic is originating (source) and to which traffic is bound (destination).
Step 10	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone)# service-policy type inspect h323-policy`	Attaches a firewall policy map to a zone-pair and enters security zone configuration mode.
Step 11	**end** **Example:** `Router(config-sec-zone)# end`	Exits security zone configuration mode and enters privileged EXEC mode.

Configuring Rate Limiting of H.323 Traffic Control Messages

Perform this task to configure a rate limit on H.323 traffic control messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect protocol-name [match-any| match-all] class-map-name**
4. **match message message-name**
5. **exit**
6. **policy-map type inspect** *protocol-name policy-map-name*
7. **class type inspect** *protocol-name class-map-name*
8. **rate-limit** *limit-number*
9. **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	**class-map type inspect protocol-name [match-any| match-all] class-map-name** **Example:** `Router(config)# class-map type inspect h323 match-any h323-ratelimit-class`	Creates a Layer 7 (application-specific) inspect type class map and enters class-map configuration mode.
Step 4	**match message message-name** **Example:** `Router(config-cmap)# match message setup`	Configures the match criterion for a class map on the basis of H.323 protocol messages.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	**policy-map type inspect** *protocol-name policy-map-name* **Example:** `Router(config)# policy-map type inspect h323 h323-ratelimit-policy`	Creates a Layer 7 inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *protocol-name class-map-name* **Example:** `Router(config-pmap)# class type inspect h323 h323-ratelimit-class`	Specifies the Layer 7 traffic (class) on which an action is to be performed and enters policy-map class configuration mode. **Note** The *class-map-name* value must match the appropriate class map name specified via the **class-map type inspect**command.
Step 8	**rate-limit** *limit-number* **Example:** `Router(config-pmap-c)# rate limit 1000`	Limits the number of messages that strike the Cisco IOS firewall every second.
Step 9	**end** **Example:** `Router(config-cmap-c)# end`	Exits policy-map class configuration mode and enters privileged EXEC mode.

Configuring Deep Packet Inspection on a Layer 3 Policy Map

Perform this task to configure deep packet inspection on a Layer 3 policy map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect policy-map-name**
4. **class type inspect class-map-name**
5. **service-policy** *protocol-name policy-map-name*
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Router> enable`	• 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 h323-policy`	Creates a Layer 3 and Layer 4 inspect type policy map.
Step 4	**class type inspect class-map-name** **Example:** `Router(config-pmap)# class type inspect h323-traffic-class`	Specifies the traffic (class) on which an action is to be performed and enters policy-map configuration mode.
Step 5	**service-policy** *protocol-name policy-map-name* **Example:** `Router(config-pmap-c)# service-policy h323 h323-ratelimit-policy`	Attaches a Layer 7 policy map to a top-level policy map.
Step 6	**end** **Example:** `Router(config-cmap-c)# end`	Exits policy-map class configuration mode and enters privileged EXEC mode.

Configuration Examples for Firewall-H.323 V3 V4 Support

Example Configuring a Voice Policy to Inspect H.323 Annex E Packets

The following example shows how to configure a voice policy to inspect the H.323 protocol Annex E packets for the “my-voice-class” class map:

```
class-map type inspect match-all my-voice-class
 match protocol h323-annexe
```

Example Configuring a H.323 Class-Map to Match Specific Messages

The following example shows how to configure an H.323 specific class map to match H.225 setup or release-complete messages only:

```
class-map type inspect h323 match-any my_h323_rt_msgs
 match message setup
 match message release-complete
```

Example Configuring a Voice Policy to Inspect H.323 Annex G Packets

The following example shows how to configure a voice policy to inspect the H.323 protocol Annex E packets for the “my-voice-class” class map:

```
class-map type inspect match-all my-voice-class
 match protocol h323-nxg
```

Example Configuring a Voice Policy to Limit Call Attempt Rate

The following example shows how to configure a voice policy to limit the call attempt rate to 16 calls per second for the calls terminated at 192.168.2.1.

```
access-list 102 permit ip any host 192.168.2.1
!
class-map type inspect match-all my_voice_class
 match protocol h323
 match access-group 102
!
class-map type inspect h323 match-any my_h323_rt_msgs
 match message setup
 policy-map type inspect h323 my_h323_policy
!
class type inspect h323 my_h323_rt_msgs
 rate-limit 16
!
policy-map type inspect my_voice_policy
 class type inspect my_voice_class
 inspect
 service-policy h323 my_h323_policy
!
```

Additional References for Firewall—H.323 V3 V4 Support

Related Documents

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

Related Topic	Document Title
Cisco security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Standards and RFCs

Standard/RFC	Title
ITU-T H.225.0	*Call signalling protocols and media stream packetization for packet-based multimedia communication systems*
ITU-T H.245	*Control protocol for multimedia communication*
ITU-T H.323 (H.323 Version 4 and earlier)	*Packet-based multimedia communications systems*
ITU-T H.450	*Supplementary services for multimedia*

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 Firewall-H.323 V3 V4 Support

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

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

Table 21: Feature Information for Firewall - H.323 V3/V4 Support

Feature Name	Releases	Feature Information
Firewall--H.323 V3/V4 Support	12.4(20)T	This feature introduces support for a range of H.323 Version 3 and Version 4 features and support for a rate-limiting mechanism to monitor call attempt rate and call aggregation. The following commands were introduced or modified: **class-map type inspect**, **class type inspect**, **match message**, **match protocol h323-annexe**, **match protocol h323-nxg**, **match protocol (zone)**, **policy-map type inspect**, **rate-limit (firewall)**, **service-policy (policy-map)**, **service-policy type inspect**.

CHAPTER 10

H.323 RAS Support

This feature introduces support for H.225 Registration, Admission, and Status (RAS) signaling in zone-based firewalls. RAS is a signaling protocol that is used between endpoints (such as gateways) and gatekeepers.

The H.225 standard is used by H.323 for call setup. H.255 includes RAS control, which is used to communicate with the gatekeeper. A RAS signaling channel enables connections between the gatekeeper and H.323 endpoints.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Restrictions for H.323 RAS Support

H.225 RAS inspection is supported only with zone-based policy firewall inspection.

How to Configure H.323 RAS Support

Configuring a Class Map for H.323 RAS Protocol Inspection

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

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 protocol** *protocol-name* [**signature**]
7. **match class-map** *class-map-name*
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	**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 class-map configuration mode.
Step 4	**match access-group** {*access-group* | **name** *access-group-name*} **Example:** `Router(config-cmap)# match access-group 101`	(Optional) Configures the match criterion for a class map based on the access control list (ACL) name or number.

	Command or Action	Purpose
Step 5	**match protocol** *protocol-name* [**signature**] **Example:** `Router(config-cmap)# match protocol h225ras`	Configures the match criterion for a class map on the basis of a specified protocol. **Note** You should specify the **h225ras** keyword to create a class-map for H.225 RAS protocol classification. For a list of supported protocols, use the command-line interface (CLI) help option (**?**) on your platform.
Step 6	**match protocol** *protocol-name* [**signature**] **Example:** `Router(config-cmap)# match protocol h323`	Configures the match criterion for a class map on the basis of a specified protocol. **Note** You should specify the **h323** keyword to create a class-map for H.323 protocol classification.
Step 7	**match class-map** *class-map-name* **Example:** `Router(config-cmap)# match class-map c1`	(Optional) Specifies a previously defined class as the match criterion for a class map.
Step 8	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.

Creating a Policy Map for H.323 RAS Protocol Inspection

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

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

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. **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 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 and enters policy-map class configuration mode.
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	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy-map configuration mode.

What to Do Next

After configuring an H.323 RAS protocol firewall policy, you want to attach the policy to a zone pair. For information on completing this task, see the "Zone-Based Policy Firewall" module.

Configuration Examples for H.323 RAS Support

Example H.323 RAS Protocol Inspection Configuration

The following example shows how to configure an H.323 RAS protocol inspection policy:

```
class-map type inspect match-any c1
 match protocol h323
 match protocol h225ras
class-map type inspect match-all c2
 match protocol icmp
!
policy-map type inspect p1
 class type inspect c1
 inspect
 class class-default
  drop
```

```
policy-map type inspect p2
 class type inspect c2
 inspect
 class class-default
  drop
!
zone security z1
 description One-Network zone
zone security z2
 description Two-Network zone
zone-pair security zp source z1 destination z2
 service-policy type inspect p1
zone-pair security zp-rev source z2 destination z1
 service-policy type inspect p2
!
interface FastEthernet1/0
 ip address 10.0.0.0 255.255.0.0
 zone-member security z1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 10.0.1.1 255.255.0.0
 zone-member security z2
 duplex auto
 speed auto
```

Example H.225 RAS Firewall Policy Configuration

The following example shows how to configure the firewall policy to inspect H.225 RAS messages:

```
interface GigabitEthernet 0/1/5
 ip address 172.16.0.0 255.255.0.0
 zone-member security private
 no shut
!
interface GigabitEthernet 0/1/6
 ip address 192.168.0.0 255.255.0.0
 zone-member security internet
 no shut
!
zone security private
zone security internet
!
class-map type inspect match-any internet-traffic-class
 match protocol h225ras
 match protocol h323
!
policy-map type inspect private-internet-policy
 class type inspect internet-traffic-class
 inspect
 class class-default
!
zone-pair security private-internet source private destination internet
 service-policy type inspect private-internet-policy
```

Additional References for H.323 RAS Support

Related Documents

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

Related Topic	Document Title
Firewall commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Technical Assistance

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

Feature Information for H.323 RAS Support

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

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

Table 22: Feature Information for H.323 RAS Support

Feature Name	Releases	Feature Information
H.323 RAS Support	12.4(11)T	H.323 RAS Support feature introduces support for H.255 Registration, Admission, and Status (RAS) signaling in zone-based firewalls. The following command was introduced or modified: **match protocol (zone)**.

CHAPTER 11

Application Inspection and Control for SMTP

The Application Inspection for SMTP feature provides an intense provisioning mechanism that can be configured to inspect packets on a granular level so that malicious network activity, related to the transfer of e-mail at the application level, can be identified and controlled. This feature qualifies the Cisco IOS firewall extended Simple Mail Transfer Protocol (ESMTP) module as an “SMTP application firewall,” which protects in a similar way to that of an HTTP application firewall.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Application Inspection and Control for SMTP

Follow the appropriate configuration tasks outlined in the Zone-Based Policy Firewall module before configuring the Application Inspection and Control for SMTP feature. This module contains important information about class-maps and policy-maps and their associated "match" statements necessary for configuring an SMTP policy.

SMTP Policy Requirements

Both SMTP and ESMTP inspection provide a basic method for exchanging e-mail messages between the client and server to negotiate capabilities and use these capabilities in an e-mail transaction. An ESMTP session is similar to an SMTP session, except for one difference--the Extended HELO (EHLO) command. The EHLO command is sent by a client to initiate the capability dialogue. After the client receives a successful response to the EHLO command, the client works the same way as SMTP, except that the client may issue new extended commands, and it may add a few parameters to the MAIL FROM and REPT TO commands.

Previously, if the Cisco IOS software was configured to inspect SMTP session only, inspection was configured by entering the **match protocol smtp** command. This action would "mask" the EHLO command to prevent capability negotiation and cause the client to go back to the HELO command and basic SMTP.

To have a workable policy for both ESMTP and SMTP inspection, the **match protocol smtp** command must be configured in the top-level policy before the Application Inspection and Control for SMTP features are implemented. See the Configuring a Default Policy for Application Inspection task for more information.

The SMTP policy (which specifies the particular SMTP configuration) is included as a child-policy in the top-level "inspect" policy-map. See the "Top-level Class Maps and Policy Maps" section in the Zone-Based Policy Firewall module for more information.

Restrictions for Application Inspection and Control for SMTP

The Application Inspection and Control for SMTP feature has the following restrictions:

- The **match cmd-line length gt** command filter can co-exist only with a **match cmd verb** command filter in the SMTP match-all class -map (**class-map type inspect smtp**). Any attempt to pair the **match cmd-line length gt** command filter with any other filter is not allowed by the CLI.
- The alternative data transfer SMTP command extension BDAT is not supported. This command is substituted for the DATA command while the SMTP body is transferred. The BDAT command extension is used by the Cisco IOS firewall to mask the CHUNKING keyword in the EHLO response to the Application Inspection and Control for SMTP feature, preventing a client from using it.
- The "mask" action can be configured only with a class having either or both of the **match cmd verb** or **match ehlo reply** commands. This action cannot be configured with a class having any other filter.

Information About Application Inspection and Control for SMTP

The Application Inspection and Control for SMTP feature inspects SMTP in a granular way and is complemented by an intensive provisioning system to help filter e-mail.

Benefits of Application Inspection and Control for SMTP

The Application Inspection and Control for SMTP feature provides the following benefits:

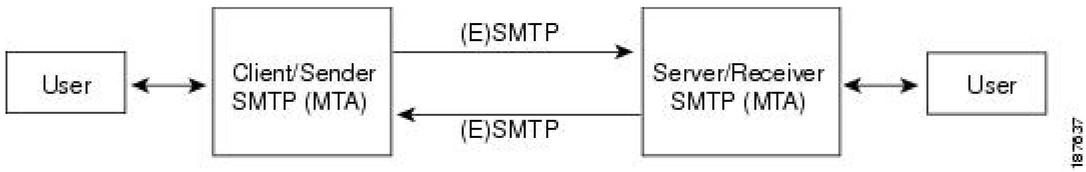
- E-mail senders and user accounts are restricted to filter spam e-mail from suspected domains.
- An action can be specified, which occurs when a number of invalid recipients appears on an SMTP connection. This action helps identify spammers who are looking for valid user accounts.
- The number of invalid SMTP recipients can be restricted by specifying a maximum number for invalid recipients on an SMTP connection.
- A pattern can be specified that identifies e-mail addressed to a particular recipient or domain in cases where a server is functioning as a relay.
- A provisioning mechanism that provides masks specified verbs in an SMTP connection to block potentially dangerous SMTP commands.
- The maximum length value for the SMTP e-mail header can be specified to prevent a Denial of Service (DoS) attack (also called a buffer overflow attack). A DoS attack occurs when the attacker continuously sends a large number of incomplete IP fragments, causing the firewall to lose time and memory while trying to reassemble the fake packets.
- The maximum length of an SMTP command line can be specified to prevent a DoS attack.
- Multipurpose Internet Mail Extension (MIME) content file-types (text, HTML, images, applications, documents,and so on) can be restricted in the body of the e-mail from being transmitted over SMTP.
- Unknown content-encoding types can be restricted from being transmitted over SMTP.
- Specified content-types and content encoding types can be restricted in the SMTP e-mail body.
- Monitor arbitrary patterns (text strings) in the SMTP e-mail message header (subject field) or body.
- A parameter in an EHLO server reply and mask can be specified to prevent a sender (client) from using the service extension in the server reply.
- An SMTP connection can be dropped with an SMTP sender (client) if the SMTP connection violates the specified policy.
- SMTP commands or the parameters returned by the server in response to an EHLO command can be explicitly masked by specifying these SMTP commands.
- An action can be logged for a class type in an SMTP policy-map.

Cisco Common Classification Policy Language

The Cisco Common Classification Policy Language (C3PL) CLI structure is used to provision ESMTP inspection. ESMTP is provisioned by defining a match criterion on an SMTP class-map and associate actions to the match criterion defined in the SMTP policy-map. The Application Inspection and Control for SMTP feature adds new match criteria and actions to the existing SMTP policy maps that are discussed in the

Zone-Based Policy Firewall module, which describes the Cisco IOS unidirectional firewall policy between groups of interfaces known as zones.

Figure 21: ESMTP Communication Between a Sender and Receiver



Common Classification Engine SMTP Database and Action Module

The Common Classification Engine (CCE) SMTP database is the site at which manually configured policy information is processed and converted into signatures. The information in these signatures is put into regular expression tables, which are then used to parse packets as they are switched by a router.

The SMTP database has two interfaces. One interface has the control plane, which is used to accept user configured policies, and the other interface has the CCE data-plane engine, which is used to classify a packet.

An action module is used as a part of the Context-Based Access Control (CBAC) SMTP inspection module to organize and trigger SMTP inspection. CBAC is used to detect and block SMTP attacks (illegal SMTP commands) and sends notifications when SMTP attacks occur.

How to Configure Application Inspection and Control for SMTP

Configuring a Default Policy for Application Inspection

If no policy is configured for SMTP, then there is no application inspection for SMTP. The firewall creates a TCP session and only performs “pinholing,” which allows an application to have access to the protected network. Having an open gap in a firewall can expose the protected system to malicious abuse. The steps below are used to provide minimum application inspection protections for SMTP by enforcing the EHLO and HELO SMTP commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match protocol smtp**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match protocol smtp** **Example:** `Router(config-cmap)# match protocol smtp`	Enables inspection for ESMTP and SMTP.

Restricting Spam from a Suspicious E-Mail Sender Address or Domain

An e-mail sender and user accounts can be restricted to filter spam e-mail from suspected domains. Spam is restricted by using the **match sender address regex** command to match the parameter-map name of a specific traffic pattern that specifies a sender domain or e-mail address in the SMTP traffic. The specified pattern is scanned in the parameter for the SMTP **MAIL FROM:** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp match-any** *class-map-name*
7. **match sender address regex** *parameter-map-name*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **log**
12. **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	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex bad-guys`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.

Step 4	**pattern** *traffic-pattern*
	
	Example:
	
	`Router(config-profile)# pattern “*deals\.com”`
	
	Example:
	
	`Router(config-profile)# pattern “*crazyperson*@wrdmail\.com”`
		Specifies the Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail.

Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp match-any c1`	Creates a class map for the SMTP protocol so the match criteria is set to match any criteria for this class map and enters class-map configuration mode.
Step 7	**match sender address regex** *parameter-map-name* **Example:** `Router(config-cmap)# match sender address regex bad-guys`	Enters the parameter-map name class, which was defined in Step 3, to specify the Cisco IOS regular expression (regex) patterns for the class-map.
Step 8	exit	Exits class-map configuration mode.

	Command or Action	Purpose
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 11	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.
Step 12	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Identifying and Restricting Spammers Searching for User Accounts in a Domain

Spammers who search for a large number of user accounts in a domain typically send the same e-mail to all the user accounts they find in this domain. Spammers can be identified and restricted from searching for user accounts in a domain by using the **match recipient count gt** command to specify an action that occurs when a number of invalid recipients appear on an SMTP connection.

Note The **match recipient count gt** command does not count the number of recipients specified in the To or Cc fields in the e-mail header.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match recipient count gt** *value*
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **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	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match recipient count gt** *value* **Example:** `Router(config-cmap)# match recipient count gt 25`	Sets a limit on the number of RCPT SMTP commands sent by the sender (client) to recipients who are specified in a single SMTP transaction. This command determines the number of RCPT lines and invalid recipients (for which the server has replied “500 No such address”) in the SMTP transaction.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode. • The *policy-map-name* argument is the name of the policy map.

	Command or Action	Purpose
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 8	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Restricting the Number of Invalid SMTP Recipients

If a sender specifies in an invalid e-mail recipient and SMTP encounters this invalid recipient on the SMTP connection, then SMTP sends an error code reply to the e-mail sender (client) to specify another recipient. In this case, the event did not violate the SMTP protocol or indicate that this particular SMTP connection is bad. However, if a pattern of invalid recipients appears, then a reasonable threshold can be set to restrict these nuisance SMTP connections. The **match recipient invalid count gt** command is used to help identify and restrict the number of invalid SMTP recipients that can appear in an e-mail from senders who try common names on a domain in the hope that they discover a valid username to whom they can send spam.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match recipient invalid count gt** *value*
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match recipient invalid count gt** *value* **Example:** `Router(config-cmap)# match recipient invalid count gt 5`	Specifies a maximum number of invalid e-mail recipients on this SMTP connection.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 8	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Specifying a Recipient Pattern to Learn Spam Senders and Domain Information

A nonexistent e-mail recipient pattern can be specified to learn about spam senders and their domain information by luring them to use this nonexistent e-mail recipient pattern. This pattern is a regular-expression (regex) that can be specified to identify an e-mail addressed to a particular recipient or domain when a server is functioning as a relay. The specified pattern is checked in the SMTP RCPT command (SMTP envelope) parameter to identify if the recipient is either used as an argument or a source-list to forward mail in the route specified in the list.

Note The **match recipient address regex** command does not operate on the To or Cc fields in the e-mail header.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp** *class-map-name*
7. **match recipient address regex** *parameter-map-name*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **log**
12. **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	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex known-unknown-users`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.

	Command or Action	Purpose
Step 4	**pattern** *traffic-pattern* **Example:** `Router(config-profile)# pattern “username@mydomain.com”`	Specifies a Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail. In the example, “username” is configured as the name for a fake e-mail account used to discover senders (and their domain) when they try to send spam e-mail to this fake account.
Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 7	**match recipient address regex** *parameter-map-name* **Example:** `Router(config-cmap)# match recipient address regex known-unknown-users`	Specifies the nonexistent e-mail recipient pattern in order to learn spam senders and their domain information by luring them to use this contrived e-mail recipient.
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 11	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.
Step 12	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Hiding Specified Private SMTP Commands on an SMTP Connection

Use this task to hide or "mask" commonly encountered SMTP verbs (SMTP commands) or specified private SMTP verbs used to provision an SMTP connection.

Specified verbs, such as the ATRN, ETRN, BDAT verbs may be considered vulnerable to exploitation if seen by a sender (client). The most commonly encountered SMTP verbs are listed along with the facility to specify a private verb as a string (using the WORD option).

Note The BDAT verb (used as an alternative to DATA) is not used, so in its place, the CHUNKING keyword is masked in the EHLO response. However, if the sender (client) continues to send the BDAT command, it is masked.

Note Using the **mask** command applies to certain **match** command filters like **match cmd verb**. Validations are performed to make this check and the configuration is not be accepted in case of invalid combinations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match cmd verb** {*verb-name* | *WORD*}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **mask**

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 smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match cmd verb** {*verb-name* | *WORD*} **Example:** `Router(config-cmap)# match cmd verb ATRN`	Specifies either the private verb name to "mask" that is used to provision an SMTP connection. • The *verb-name* argument is the name of an SNMP command verb. • The *WORD* argument is the name of a user-specified SMTP command verb, which is treated as an unknown verb and is masked regardless of whether the 'mask action is configured for the class or not.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 8	**mask** **Example:** `Router(config-pmap)# mask`	Explicitly masks the specified SMTP commands or the parameters returned by the server in response to an EHLO command.

Preventing a DoS Attack by Limiting the Length of the SMTP Header

A DoS attack (also called a buffer overflow attack) by a malicious sender (client) can cause the SMTP application firewall to lose time and memory while trying to reassemble the fake packets (large e-mail headers) associated with the e-mail. In an SMTP transaction, the header portion of an e-mail is considered part of the DATA area, which contains fields like Subject, From, To, Cc, Date, and proprietary information, which is used by a recipient's e-mail agent to process the e-mail. A DoS attack can be prevented by using the **match header length gt** command to limit the length of the SMTP header that can be received. If a match is found, possible actions that can be specified within the policy are as follows: allow, reset, or log (the log action triggers a syslog message when a match is found).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match header length gt** *bytes*
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **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	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match header length gt** *bytes* **Example:** `Router(config-cmap)# match header length gt 16000`	Specifies a value from 1 to 65535 that limits the maximum length of the SMTP header in bytes to thwart DoS attacks.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.

	Command or Action	Purpose
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 8	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Preventing a DoS Attack by Limiting the Length or TYPE of SMTP Command Line

The following task is used to limit the length of an SMTP command line to prevent a DoS attack, which occurs when a malicious sender (client) specifies large command lines in an e-mail to perform DoS attacks on SMTP servers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name*}
4. **match cmd** {**line length gt** *length* | **verb** {**AUTH** | **DATA** | **EHLO** | **ETRN** | **EXPN** | **HELO** | **HELP** | **MAIL NOOP** | **QUIT** | **RCPT** | **RSET** | **SAML** | **SEND** | **SOML** | **STARTTLS** | **VERB** | **VRFY** | **WORD**}}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map. **Note** If no **match cmd verb** command statement is specified in a **class-map type inspect smtp match-all** command statement for a class-map, which contains the **match cmd line length gt** command statement, then the class-map applies to all SMTP commands.
Step 4	**match cmd** {**line length gt** *length* | **verb** {**AUTH** | **DATA** | **EHLO** | **ETRN** | **EXPN** | **HELO** | **HELP** | **MAIL NOOP** | **QUIT** | **RCPT** | **RSET** | **SAML** | **SEND** | **SOML** | **STARTTLS** | **VERB** | **VRFY** | **WORD**}} **Example:** `Router(config-cmap)# match header length gt 16000`	Specifies a value that limits the length of the ESMTP command line or ESMTP command line verb used to thwart DoS attacks. • The *length* argument specifies the ESMTP command line greater than the length of a number of characters from 1 to 65535.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.

	Command or Action	Purpose
Step 8	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Examples

The following configuration has class-map c2 match when the length of the e-mail (MAIL) command exceeds 256 bytes.

When the **class-map type inspect smtp match-all** command statement is configured with the **match cmd verb** command statement, only the **match cmd line length gt** command statement can coexist.

```
class-map type inspect smtp match-all c2
  match cmd line length gt 256
  match cmd verb MAIL
```

There are no match restrictions in case of a **class-map type inspect smtp match-any** command statement for a class map because the class-map applies to all SMTP commands.

Restricting Content File Types in the Body of the E-Mail

The **match mime content-type regex** command is used to specify MIME content file types, which are restricted in attachments in the body of the e-mail being sent over SMTP. See the Example: MIME E-Mail Format section for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name*}
7. **match mime content-type regex** *content-type-regex*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **log**

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 regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex jpeg`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.
Step 4	**pattern** *traffic-pattern* **Example:** `Router(config-profile)# pattern “*image//*”`	Specifies a Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail.
Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.
Step 7	**match mime content-type regex** *content-type-regex* **Example:** `Router(config-cmap)# match mime content-type regex jpeg`	Specifies the MIME content file type, which are restricted in attachments in the body of the e-mail being sent over SMTP. • The *content-type-regex* argument is the type of content in the MIME header in regular expression form. This example lets the user specify any form of JPEG image content to be restricted.

	Command or Action	Purpose
		Note The actual content of the MIME part is not checked to see if it matches with the declared content-type in the MIME header.
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 11	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

Restricting Unknown Content Encoding Types from Being Transmitted

Unknown MIME content-encoding types or values can be restricted from being transmitted over SMTP by using one of the following parameters with the **match mime encoding**command.

These preconfigured content-transfer-encoding types act as a filter on the content-transfer-encoding field in the MIME header within the SMTP body. The uuencode encoding type is not recognized as a standard type by the MIME RFCs because many subtle differences exist in its various implementations. However, since it is used by some mail systems, the **x-uuencode** type is included in the preconfigured list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
4. **match mime encoding** {**unknown** | *WORD* | *encoding-type*}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **log**

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** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.
Step 4	**match mime encoding** {**unknown** | *WORD* | *encoding-type*} **Example:** `Router (config-cmap)# match mime encoding quoted-printable`	Restricts unknown MIME content-encoding types or values. • The **unknown** keyword is used if content-transfer-encoding value in the e-mail does not match any of the ones in the list to restrict unknown and potentially dangerous encodings. • The *WORD* argument is a user-defined content-transfer encoding type, which must begin with "X-" (for example, "X-myencoding-scheme"). • The *encoding-type* argument specifies one of the following preconfigured content-transfer-encoding types: • **7-bit**-ASCII characters

	Command or Action	Purpose
		• **8-bit**-Facilitates the exchange of e-mail messages containing octets outside the 7-bit ASCII range. • **base64**-Any similar encoding scheme that encodes binary data by treating it numerically and translating it into a base 64 representation. • **quoted-printable**-Encoding using printable characters (that is alphanumeric and the equals sign "=") to transmit 8-bit data over a 7-bit data path. It is defined as a MIME content transfer encoding for use in Internet e-mail. • **binary**-Representation for numbers using only two digits (usually, 0 and 1). • **x-uuencode**-Nonstandard encoding. **Note** The **quoted-printable** and **base64** encoding types tell the e-mail client that a binary-to-text encoding scheme was used and that appropriate initial decoding is necessary before the message can be read with its original encoding.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 8	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

Specifying a Text String to Be Matched and Restricted in the Body of an E-Mail

The **match body regex** command can be used to specify an arbitrary text expression to restrict specified content-types and content encoding types for text and HTML in the body of the e-mail. The text or HTML pattern is scanned only if the encoding is 7-bit or 8-bit and the encoding is checked before attempting to match the pattern. If the pattern is of another encoding type (for example, base64, zip files, and so on), then the pattern cannot be scanned.

Note Using this command can impact performance because the complete SMTP connection has to be scanned.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
7. **match body regex** *parameter-map-name*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **log**

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 regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex doc-data`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.

	Command or Action	Purpose
Step 4	**pattern** *traffic-pattern* **Example:** `Router(config-profile)# pattern “*UD-421590*”`	Specifies a Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail.
Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.
Step 7	**match body regex** *parameter-map-name* **Example:** `Router(config-cmap)# match body regex doc-data`	Specifies an arbitrary text expression to restrict specified content-types and content encoding types for text and HTML in the “body” of the e-mail.
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 11	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

Configuring the Monitoring of Text Patterns in an SMTP E-Mail Subject Field

The **match header regex** command can be used specify an arbitrary text expression in the SMTP e-mail message header (Subject field) or e-mail body such as Subject, Received, To, or other private header fields to monitor text patterns.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name*}
7. **match header regex** *parameter-map-name*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **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	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex lottery-spam`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.

	Command or Action	Purpose
Step 4	**pattern** *traffic-pattern* **Example:** `Router(config-profile)# pattern "Subject:*lottery*"`	Specifies a Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail.
Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.
Step 7	**match header regex** *parameter-map-name* **Example:** `Router(config-cmap)# match header regex lottery-spam`	Specifies an arbitrary text expression in the SMTP e-mail message header to monitor text patterns.
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 11	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Configuring a Parameter to Be Identified and Masked in the EHLO Server Reply

The **match reply ehlo** command is used to identify and mask a service extension parameter in the EHLO server reply (for example, 8BITMIME and ETRN) to prevent a sender (client) from using that particular service extension.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
4. **match reply ehlo** {*parameter* | *WORD*}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **log**
9. **mask**

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** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.

	Command or Action	Purpose
Step 4	**match reply ehlo** {*parameter* | *WORD*} **Example:** `Router(config-cmap)# match reply ehlo ETRN`	Identifies and masks a service extension parameter in the EHLO server reply. • The *parameter* argument specifies a parameter from the well-known EHLO keywords. • The *WORD* argument specifies an extension which is not on the EHLO list.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 8	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.
Step 9	**mask** **Example:** `Router(config-pmap)# mask`	Explicitly masks the specified SMTP commands or the parameters returned by the server in response to an EHLO command.

Configuring a Logging Action for a Class Type in an SMTP Policy-Map

A logging action can be configured for a class type in an SMTP policy-map when conditions specified by the traffic class are met. The logging action results in a LOG_WARNING syslog message followed by the specific log message. The log message format is similar to other application firewall modules (for example, HTTP, IM, Peer-to-Peer (P2P)); session initiator/responder information, and zone-pair and class names.

Note The log action currently exists for other types of policy-maps (http, pop3).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
4. **match cmd verb** {*parameter* | *WORD*}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **log**

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** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.
Step 4	**match cmd verb** {*parameter* | *WORD*} **Example:** `Router(config-cmap)# match cmd verb ATRN`	Identifies and masks a service extension parameter in the EHLO server reply. • The *parameter* argument specifies a parameter from the well-known EHLO keywords. • The *WORD* argument specifies an extension which is not on the EHLO list.
Step 5	exit	Exits class-map configuration mode.

	Command or Action	Purpose
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 8	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

Configuration Examples for Application Inspection and Control for SMTP

Example Creating a Pinhole for the SMTP Port

The following example shows a configuration without any Layer 7 SMTP policy that creates a pinhole only for the SMTP port. Any command sent to the server, including the EHLO command is accepted.

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

Note No SMTP policy is configured by default. If an SMTP policy is not configured, then no SMTP inspection is done by default.

Example Preventing ESMTP Inspection

If a user decides to create a workable policy that is configured for SMTP inspection only, then it now needs to be explicitly specified in the policy.

The following example can be used to prevent ESMTP inspection:

```
class-map type inspect smtp c1
 match cmd verb EHLO
policy-map type inspect smtp c1
 class type inspect smtp c1
  mask
```

Example MIME E-Mail Format

The format of data being transmitted through SMTP is specified by using the MIME standard, which uses headers to specify the content-type, encoding, and the filenames of data being sent (text, html, images, applications, documents and so on). The following is an example of an e-mail using the MIME format:

```
From: "username2" <username2@example.com>
To: username3 <username3@example.com>
Subject: testmail
Date: Sat, 7 Jan 2006 20:18:47 -0400
Message-ID: <000dadf7453e$bee1bb00$8a22f340@oemcomputer>
MIME-Version: 1.0
Content-Type: image/jpeg;
name='picture.jpg'
Content-Transfer-Encoding: base64
```

<base64 encoded data for the picture.jpg image>

In the above example, the "name='picture.jpg'" is optional. Even without the definition, the image is sent to the recipient. The e-mail client of the recipient may display the image as "part-1" or "attach-1" or it may render the image in-line. Also, attachments are not 'stripped' from the e-mail. If a content-type for which reset action was configured is detected, an 5XX error code is sent and the connection is closed, in order to prevent the whole e-mail from being delivered. However, the remainder of the e-mail message is sent.

Additional References for Application Inspection and Control for SMTP

Related Documents

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

Related Topic	Document Title
ESMTP firewall information.	ESMTP Support for Cisco IOS Firewall
Information for configuring an SMTP policy.	Zone-Based Policy Firewall

Standards and RFCs

Standard/RFC	Title
RFC 1869 and other SMTP RFC extensions apart from RFC 821	*SMTP Service Extensions*

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 Application Inspection and Control for SMTP

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

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

Table 23: Feature Information for Application Inspection and Control for SMTP

Feature Name	Releases	Feature Information
Application Inspection and Control for SMTP	12.4(20)T	The Application Inspection and Control for SMTP feature provides an intense provisioning mechanism that can be configured to inspect packets on a granular level so that malicious network activity, related to the transfer of e-mail at the application level, can be identified and controlled. This feature qualifies the Cisco IOS firewall extended SMTP (ESMTP) module as an "SMTP application firewall," which protects in a similar way to that of an HTTP application firewall. The following commands were introduced or modified by this feature: **log (policy-map and class-map) , mask (policy-map)**, **match body regex**, **match cmd**, **match header length gt**, **match header regex**, **match mime content-type regex**, **match mime encoding**, **match sender address regex**, **match recipient address regex**, **match recipient count gt**, **match recipient invalid count gt**, **match reply ehlo**, **reset (policy-map)**.

Glossary

C3PL --Cisco Common Classification Policy Language. Structured, feature-specific configuration commands that use policy maps and class maps to create traffic policies based on events, conditions, and actions.

EHLO --Extended HELO substitute command for starting the capability negotiation. This command identifies the sender (client) connecting to the remote SMTP server by using the ESMTP protocol.

ESMTP --Extended Simple Mail Transfer Protocol. Extended version of the Simple Mail Transfer Protocol (SMTP), which includes additional functionality, such as delivery notification and session delivery. ESMTP is described in RFC 1869, SMTP Service Extensions.

HELO --Command that starts the SMTP capability negotiation. This command identifies the sender (client) connecting to the remote SMTP server by its fully qualified DNS hostname.

MAIL FROM --Start of an e-mail message that identifies the sender e-mail address (and name, if used), which appears in the From: field of the message.

MIME --Multipurpose Internet Mail Extension. Standard for transmitting nontext data (or data that cannot be represented in plain ASCII code) in e-mail, such as binary, foreign language text (such as Russian or Chinese), audio, or video data. MIME is defined in RFC 2045.

RCPT TO --Recipient e-mail address (and name, if used) that can be repeated multiple times for a likely message to deliver a single message to multiple recipients.

SMTP --Simple Mail Transfer Protocol. Internet protocol providing e-mail services.

CHAPTER 12

Subscription-Based Cisco IOS Content Filtering

The Subscription-based Cisco IOS Content Filtering feature interacts with the Trend Micro URL filtering service so that HTTP requests can be allowed or blocked, and logged, based on a content filtering policy. The content filtering policy specifies how to handle items such as web categories, reputations (or security ratings), trusted domains, untrusted domains, and keywords. URLs are cached on the router, so that subsequent requests for the same URL do not require a lookup request, thus improving performance.

Support for third-party URL filtering servers SmartFilter (previously N2H2) and Websense, which was introduced with Cisco IOS Release 12.2(11)YU and integrated into Cisco IOS Release 12.2(15)T, continues to be available.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Subscription-Based Cisco IOS Content Filtering

Cisco IOS Firewalls and Zone-Based Policy Firewall

You should have an understanding of how to configure Cisco IOS firewalls and understand the concepts of traffic filtering, traffic inspection, and zone-based policy.

Trend Micro Requirements

Before you can configure the Subscription-Based Cisco IOS Content Filtering feature on the router, you must:

- Purchase the Cisco IOS Content Filtering Subscription Service from Cisco.
- Receive the Product Authorization Key (PAK) in the mail.
- Activate your license at www.cisco.com/go/license . You will need the serial number for the router and the PAK.
- Download and install the security certificate as described here:

Install Trusted Authority Certificates on Cisco IOS Routers for Trend URL Filtering Support

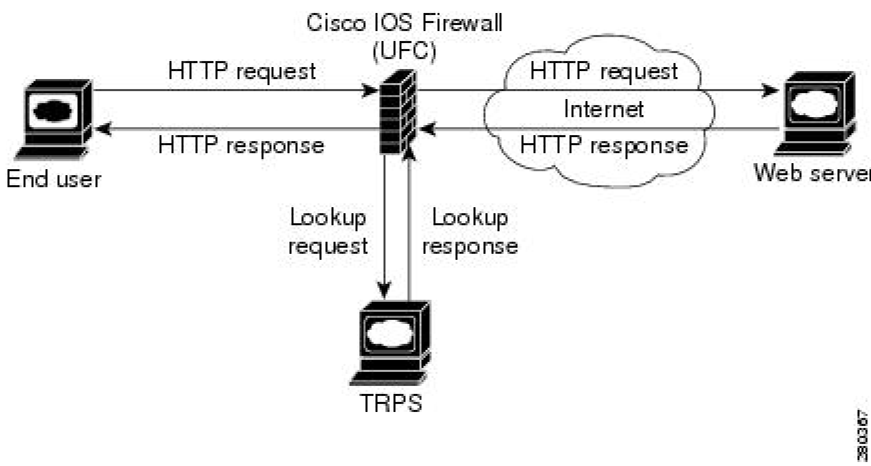
- Use the **trm register** command in privileged EXEC mode to register the router with the Trend Router Provisioning Server (TRPS).

Information About Subscription-Based Cisco IOS Content Filtering

Overview of Subscription-Based Cisco IOS Content Filtering

The Subscription-Based Cisco IOS Content Filtering service interacts with the Trend Micro filtering service URL requests based on URL filtering policy. The figure below and the following steps provide a brief overview of Cisco IOS content filtering.

Figure 22: Subscription-Based Cisco IOS Content Filtering Sample Topology



1 The end user opens a web browser and browses to a web page.

2 The browser sends an HTTP request to the Cisco IOS content filtering service.

3 The Cisco IOS content filtering service receives the request, forwards the request to the web server while simultaneously extracting the URL and sending a lookup request to the TRPS.

4 The TRPS receives the lookup request and retrieves the URL category for the requested URL from its database.

5 The TRPS sends the lookup response to the Cisco IOS content filtering service.

6 The Cisco IOS content filtering service receives the lookup response and permits or denies the URL as specified by a Trend Micro URL filtering policy on the router.

7 The Cisco IOS content filtering service caches the URL and lookup response.

Overview of URL Filtering Policies

A URL filtering policy contains an association of classes and actions and a set of URL filtering parameters that specify how the system handles URL requests.

- A class is a set of match criteria that identifies traffic based on its content. Classes are specified by class maps.
- An action is a specific function associated with a given traffic class. For URL traffic, the actions include **allow**, **log**, and **reset**.
- Classes and actions are associated with one another in a policy map.
- URL filtering parameters specify information about the URL filtering server. URL filtering parameters are specified in a parameter map.
- A URL filtering policy goes into effect when it is attached to a zone pair with the service-policy command.
- You can configure multiple URL filtering policies on the system.

Cisco IOS Content Filtering Modes

Subscription-based Cisco IOS content filtering operates in one of three modes: local filtering mode, URL database filtering mode, and allow mode.

Local Filtering Mode

In this mode, the Cisco IOS content filtering service first tries to match the requested URL with the local lists of trusted domains (white list), untrusted domains (black list), and blocked keywords. If a match is not found, the Cisco IOS content filtering service forwards the lookup request to the URL filtering server as specified in the policy. If the Cisco IOS content filtering service cannot establish communication with the URL filtering server, the system enters allow mode.

The system is in local filtering mode when a URL filtering policy for a URL filtering server has not been specified and when the system cannot establish a connection with the URL filtering server.

URL Database Filtering Mode

In this mode, the Cisco IOS content filtering service has connectivity with the URL filtering server; it can send URL lookup requests to and receive URL lookup responses from the URL filtering server.

In the case of a TRPS, the Cisco IOS content filtering service sends a URL category lookup request to the TRPS and the TRPS responds with the URL category and the URL reputation. Based on the policy set for the URL category and reputation, the HTTP request is allowed, denied, or logged. If a policy has not been configured for the URL category or reputation, the default is to permit the HTTP response.

In the case of SmartFilter and Websense servers, the Cisco IOS content filtering service sends a URL lookup request to the URL database server and the server responds with either a permit or deny message. URL filtering policies for SmartFilter and Websense servers specify a server-based action.

Allow Mode

When the Cisco IOS content filtering service is unable to communicate with the URL filtering server, the system enters allow mode. The default setting for allow mode is off, and all HTTP requests that pass through local filtering mode are blocked. When allow mode is on, all HTTP requests that passed through local filtering mode are allowed.

When both local filtering and URL database filtering modes fail, the system goes into allow mode. If the allow mode action is set to on, all URL requests are allowed. Otherwise, all HTTP requests are blocked.

Benefits of Subscription-Based Cisco IOS Content Filtering

The Subscription-Based Cisco IOS Content Filtering feature allows you to control web traffic based on a particular policy. This following sections describe available with this feature:

- Benefits of Subscription-Based Cisco IOS Content Filtering, on page 293
- Benefits of Subscription-Based Cisco IOS Content Filtering, on page 293
- Benefits of Subscription-Based Cisco IOS Content Filtering, on page 293

White Lists, Black Lists, and Blocked Keyword Lists

This function, which supports the local filtering mode, provides a means of specifying per-policy lists of trusted domain names (white lists), untrusted domain names (black lists), and URL keywords to be blocked (blocked keywords).

When the domain name in a URL request matches an item on the white list, the Cisco IOS content filtering service sends the URL response to the end user's browser directly without sending a lookup request to the TRPS. When the domain name in a URL request matches an item on the black list, the Cisco IOS content filtering service blocks the URL response to the end user's browser. You can specify complete domain names or use the wildcard character * to specify partial domain names.

When a URL contains a keyword, the Cisco IOS content filtering service blocks the URL response directly without sending a lookup request to the URL filtering server. The content filtering service looks at the content of the URL beyond the domain name when making keyword comparisons. For example, if the keyword list contains the word "example," the URL "www.example1.com/example" matches on the keyword example, whereas the URL "www.example.com/example1' does not. You can specify complete words or use the wildcard character * to specify a word pattern.

Caching Recent Requests

This function provides a cache table that contains information about the most recently requested URLs. As a result, a subsequent request for the same URL can be handled by the system without sending a lookup request to the URL filtering server, thus keeping response time to a minimum. In the case of a Trend Micro filtering server, the cache table includes category information for the requested URL. In the case of SmartFilter and Websense filtering servers, the cache table specifies whether the requested URL is allowed or denied.

You can configure the size of the cache table and the length of time an entry remains in the cache table before it expires.

Packet Buffering

This buffering scheme allows the Cisco IOS content filtering service to store HTTP responses while waiting for the URL lookup response from the URL filtering server. The responses remain in the buffer until the response is received from the URL filtering server. If the response indicates that the URL is allowed, the content filtering service releases the HTTP response in the buffer to the end user's browser; if the status indicates that the URL is blocked, the content filtering service discards the HTTP responses in the buffer and closes the connection to both ends. This function prevents numerous HTTP responses from overwhelming your system.

You can specify the number of responses that can be held in the buffer. The default is 200.

Support for SmartFilter and Websense URL Filtering Servers

The Cisco IOS content filtering service provides support for SmartFilter and Websense URL filtering servers. In the case of these third-party URL filtering servers, you configure the URL filtering policy on the router to perform the action specified by the URL filtering server--that is, to allow or deny access to the requested URL.

How to Configure Subscription-Based Cisco IOS Content Filtering

Configuring Class Maps for Local URL Filtering

The Cisco IOS content filtering service filters URL requests on the basis of match criteria in class maps. To enable local URL filtering, you must specify at least one class map each for trusted domains, untrusted domains, and blocked keywords. The match criteria for these class maps are specified in a parameter map, which must be configured before the class map is configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type urlf-glob** *parameter-map-name*
4. **pattern** *expression*
5. **exit**
6. Repeat Steps 3 through 5 twice.
7. **class-map type urlfilter match-any** *class-map-name*
8. **match server-domain urlf-glob** *parameter-map-name*
9. **exit**
10. Repeat Step 7 through Step 9.
11. **class-map type urlfilter match-any** *class-map-name*
12. **match url-keyword urlf-glob** *parameter-map-name*
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**parameter-map type urlf-glob** *parameter-map-name*
	
	Example:
	
	`Router(config)#`
	`parameter-map type urlf-glob trusted-domain-param`
		Creates the parameter map for trusted domains and enters profile configuration mode.

Step 4	**pattern** *expression* **Example:** `Router(config-profile)# pattern www.example.com`	Specifies the matching criteria in the parameter map.
Step 5	**exit** **Example:** `Router(config-profile)# exit`	Returns to global configuration mode.
Step 6	Repeat Steps 3 through 5 twice.	Configures the remaining two parameter maps required for local URL filtering: one for untrusted domains and one for URL keywords.

Step 7	**class-map type urlfilter match-any** *class-map-name*
	
	Example:
	
	`Router(config)# class-map type urlfilter`
	`match-any trusted-domain-class`
		Creates a URL filter class for trusted domains and enters class map configuration mode.

Step 8	**match server-domain urlf-glob** *parameter-map-name*
	
	Example:
	
	`Router(config-cmap)# match server-domain`
	`urlf-glob trusted-domain-param`
		Configures the matching criteria for the trusted domain class map.

Step 9	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 10	Repeat Step 7 through Step 9.	Creates and configures the class map for untrusted domains and returns to global configuration mode.
Step 11	**class-map type urlfilter match-any** *class-map-name* **Example:** `Router(config)# class-map type urlfilter match-any keyword-class`	Creates the class map for URL keywords and enters class map configuration mode.
Step 12	**match url-keyword urlf-glob** *parameter-map-name* **Example:** `Router(config-cmap)# match url-keyword urlf-glob keyword-param`	Configures the match criteria for the URL keyword class map based on the previously configured parameter map.
Step 13	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.

Configuring Class Maps for Trend Micro URL Filtering

To enable Trend Micro URL filtering, you must configure one or more class maps that specify the match criteria for URL categories. As an option, you can configure one or more class match that specify match criteria for URL reputations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type urlfilter trend** [**match-any**] *class-map-name*
4. **match url category** *category-name*
5. Repeat Step 4 until all categories for the class map have been specified.
6. **exit**
7. Repeat Steps 3 through 6 until all classes for Trend Micro URL category filtering have been configured.
8. **class-map type urlfilter trend** [**match-any**] *class-map-name*
9. **match url reputation** *reputation-name*
10. Repeat Step 9 until all reputations for the class map have been specified.
11. **exit**
12. Repeat Steps 8 through 11 until all classes for Trend Micro URL reputation filtering have been configured.

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 urlfilter trend** [**match-any**] *class-map-name* **Example:** `Router(config)# class-map type urlfilter trend match-any drop-category`	Creates a class map for Trend Micro URL category filtering and enters class map configuration mode.
Step 4	**match url category** *category-name* **Example:** `Router(config-cmap)# match url category Gambling`	Specifies the matching criteria for the Trend Micro URL filtering class.
Step 5	Repeat Step 4 until all categories for the class map have been specified.	(Optional) Specifies additional matching criteria.
Step 6	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.
Step 7	Repeat Steps 3 through 6 until all classes for Trend Micro URL category filtering have been configured.	(Optional) Configures additional classes for URL filtering.
Step 8	**class-map type urlfilter trend** [**match-any**] *class-map-name* **Example:** `Router(config)# class-map type urlfilter trend match-any drop-reputation`	(Optional) Creates a class map for Trend Micro URL reputation filtering and enters class map configuration mode.
Step 9	**match url reputation** *reputation-name* **Example:** `Router(config-cmap)# match url reputation PHISHING`	(Optional) Specifies the matching criteria for the Trend Micro URL filtering class.

	Command or Action	Purpose
Step 10	Repeat Step 9 until all reputations for the class map have been specified.	(Optional) Specifies additional matching criteria.
Step 11	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.
Step 12	Repeat Steps 8 through 11 until all classes for Trend Micro URL reputation filtering have been configured.	(Optional) Configures additional classes for URL filtering.

Configuring Parameter Maps for Trend Micro URL Filtering

To enable Trend Micro URL filtering, you must configure the global parameters for the TRPS in a parameter map. You can configure only one global Trend Micro parameter map. As an option, you can configure per-policy TRPS parameters in a per-policy parameter map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type trend-global** *parameter-map-name*
4. **server** {*server-name* | *ip-address*} [**http-port** *port-number*] [**https-port** *port-number*] [**retrans** *retransmission-count*] [**timeout** *seconds*]
5. **alert** {**on** | **off**}
6. **cache-entry-lifetime** *hours*
7. **cache-size maximum-memory** *kilobyte*
8. **exit**
9. **parameter-map type urlfpolicy trend** *parameter-map-name*
10. **allow-mode** {**on** | **off**}
11. **block-page** {**message** *string* | **redirect-url** *url*}
12. **max-request** *number-requests*
13. **max-resp-pak** *number-responses*
14. **truncate hostname**
15. **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 trend-global** *parameter-map-name* **Example:** `Router(config)# parameter-map type trend-global global-trend param`	Creates the parameter map for global parameters for the TRPS and enters profile configuration mode.
Step 4	**server** {*server-name* | *ip-address*} [**http-port** *port-number*] [**https-port** *port-number*] [**retrans** *retransmission-count*] [**timeout** *seconds*] **Example:** `Router(config-profile)# server trps1.trendmicro.com retrans 5 timeout 200`	(Optional) Configures basic server parameters for the TRPS.
Step 5	**alert** {**on** | **off**} **Example:** `Router(config-profile)# alert on`	(Optional) Turns on or off URL-filtering server alert messages that are displayed on the console.
Step 6	**cache-entry-lifetime** *hours* **Example:** `Router(config-profile)# cache-entry-lifetime 3`	(Optional) Specifies how long, in hours, an entry remains in the cache table.
Step 7	**cache-size maximum-memory** *kilobyte* **Example:** `Router(config-profile)# cache-size maximum-memory 512`	(Optional) Configures the size of the categorization cache.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config)# exit`	Returns to global configuration mode.
Step 9	**parameter-map type urlfpolicy trend** *parameter-map-name* **Example:** `Router(config)# parameter-map type urlfpolicy trend trend-param-map`	(Optional) Creates a parameter map for the per-policy parameters for a Trend Micro URL filtering policy and enters profile configuration mode.
Step 10	**allow-mode** {**on** | **off**} **Example:** `Router(config-profile)# allow-mode on`	(Optional) Specifies whether to allow or block URL requests when the URL filtering process does not have connectivity to the specified URL filtering service. • When allow mode is **on**, all unmatched URL requests are allowed. • When allow mode is **off**, all unmatched URL requests are blocked. • The default is **off**.
Step 11	**block-page** {**message** *string* | **redirect-url** *url*} **Example:** `Router(config-profile)# block-page message "This page is blocked by Trend policy."`	(Optional) Specifies the response to a blocked URL request. • **message** *string* --Specifies the message text to be displayed when a URL request is blocked. • **redirect-url** *url* --Specifies the URL of the web page to be displayed when a URL request is blocked.
Step 12	**max-request** *number-requests* **Example:** `Router(config-profile)# max-request 5000`	(Optional) Specifies the maximum number of pending URL requests. • The range is from 1 to 2147483647. • The default is 1000.
Step 13	**max-resp-pak** *number-responses* **Example:** `Router(config-profile)# max-resp-pak 500`	(Optional) Specifies the number of HTTP responses that can be buffered. • The range is from 0 to 20000. • The default is 200.

	Command or Action	Purpose
Step 14	**truncate hostname** **Example:** `Router(config-profile)# truncate hostname`	(Optional) Specifies that URLs be truncated at the end of the domain name.
Step 15	**exit** **Example:** `Router(config-profile)# exit`	Returns to global configuration mode.

Configuring URL Filtering Policies

URL filtering policies are configured by associating classes with actions and specifying the URL filtering parameters for the URL filtering server. To enable subscription-based Cisco IOS content filtering, you must configure a Trend Micro URL filtering policy. To enable SmartFilter or Websense URL filtering, you must configure a SmartFilter or Websense URL filtering policy.

Before You Begin

Before you can configure a URL filter policy, you must have previously configured the URL filter classes to which the policy applies and have specified a parameter map for the filtering server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect urlfilter** *policy-map-name*
4. **parameter type urlfpolicy** [**local** | **trend** | **n2h2** | **websense**] *parameter-map-name*
5. **class type urlfilter** [**trend** | **n2h2** | **websense**] *class-map-name*
6. **allow** | **reset** | **server-specified-action**
7. **log**
8. **exit**
9. Repeat Steps 4 through 8 for the remaining classes of traffic to which the policy applies.
10. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect urlfilter** *policy-map-name* **Example:** `Router(config)# policy-map type inspect urlfilter trend-policy`	Creates the policy map for the URL filtering policy and enters policy-map configuration mode.
Step 4	**parameter type urlfpolicy** [**local** | **trend** | **n2h2** | **websense**] *parameter-map-name* **Example:** `Router(config-pmap)# parameter type urlfpolicy trend trend-parameters`	Specifies the parameters in a parameter map for the URL filtering server.
Step 5	**class type urlfilter** [**trend** | **n2h2** | **websense**] *class-map-name* **Example:** `Router(config-pmap)# class type urlfilter trusted-domain-class`	Specifies the class to which the policy applies and enters policy-map class configuration mode.
Step 6	**allow** | **reset** | **server-specified-action** **Example:** `Router(config-pmap-c)# allow`	Specify the action to take: • **allow** --Allows traffic matching the pattern specified by the class. • **reset** --Blocks traffic matching the pattern specified by the class by resetting the connection on both ends. • **server-specified-action** --Allows or blocks traffic as specified by the URL filtering server.
Step 7	**log** **Example:** `Router(config-pmap-c)# log`	(Optional) Logs the request for traffic matching the pattern specified by the class.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy map configuration mode.
Step 9	Repeat Steps 4 through 8 for the remaining classes of traffic to which the policy applies.	(Optional) Specifies additional classes and actions for the policy
Step 10	**exit** **Example:** `Router(config-pmap)# exit`	Returns to global configuration mode.

Attaching a URL Filtering Policy

After you have configured a URL filtering policy, you attach the policy to an inspect type policy map that defines the traffic to be inspected and the actions to be taken based on the characteristics of the traffic. Then, you attach the inspect type policy map as a service policy to a particular target (zone-pair). After you attach the policy, you must configure the interfaces that belong to the zone. See the *Cisco IOS Security Configuration Guide* for more information.

Before You Begin

If you do not want to use the default parameters for inspecting traffic, use the **parameter-map type inspect** command to configure the parameters related to the inspect action.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-all** *class-map-name*
4. **match protocol http**
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect** *parameter-map-name*
9. **service-policy urlfilter** *policy-map-name*
10. **exit**
11. **class class-default**
12. **drop**
13. **exit**
14. **exit**
15. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**} **destination** [**self** | *destination-zone-name*]
16. **service-policy type inspect** *policy-map-name*
17. **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	**class-map type inspect match-all** *class-map-name* **Example:** `Router(config)# class-map type inspect match-all http-class`	Creates an inspect type class map and enters class map configuration mode.

	Command or Action	Purpose
Step 4	**match protocol http** **Example:** `Router(config-cmap)# match protocol http`	Specifies the HTTP protocol as the match criteria for the class map.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect trend-global-policy`	Creates an inspect type policy map and enters policy-map configuration mode. This policy map defines the traffic to be inspected and the actions to take on that traffic.
Step 7	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect http-class`	Specifies the HTTP traffic class to be inspected by the policy and enters policy-map class configuration mode.
Step 8	**inspect** *parameter-map-name* **Example:** `Router(config-pmap-c)# inspect global`	Specifies the inspect action on HTTP traffic.
Step 9	**service-policy urlfilter** *policy-map-name* **Example:** `Router(config-pmap-c)# service-policy urlfilter trend-policy`	Attaches the URL filter policy to all HTTP traffic.
Step 10	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy-map configuration mode.
Step 11	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Creates the default class--that is, all traffic that does not match the criteria specified by the HTTP class map--and enters policy-map class configuration mode.

	Command or Action	Purpose
Step 12	**drop** **Example:** `Router(config-pmap-c)# drop`	Specifies the action to take on traffic in the default class--that is, to drop all non-HTTP traffic.
Step 13	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy-map configuration mode.
Step 14	**exit** **Example:** `Router(config-pmap)# exit`	Returns to global configuration mode.
Step 15	**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 16	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect trend-policy`	Attaches a URL filtering policy to the destination zone pair.
Step 17	**exit** **Example:** `Router(config-sec-zone-pair)# exit`	Returns to global configuration mode.

Configuration Examples for Cisco IOS Content Filtering

Example Configuring Class Maps for Local URL Filtering

The following example shows class maps for trusted domains, untrusted domains, and URL keywords. The required parameter maps are configured first.

```
parameter-map type urlf-glob trusted-domain-param
 pattern www.example1.com
 pattern *.example2.com
!
parameter-map type urlf-glob untrusted-domain-param
 pattern www.example3.com
 pattern www.example4.com
!
parameter-map type urlf-glob keyword-param
 pattern mp3
 pattern jobs
class-map type urlfilter match-any untrusted-domain-class
 match server-domain urlf-glob untrusted-domain-param
class-map type urlfilter match-any trusted-domain-class
 match server-domain urlf-glob trusted-domain-param
class-map type urlfilter match-any keyword-class
 match url-keyword urlf-glob keyword-param
```

Example Configuring Class Maps for Trend Micro URL Filtering

The following example shows a class map that defines the class drop-category, which specifies traffic that matches the defined URL categories:

```
class-map type urlfilter trend match-any drop-category
 match url category Gambling
 match url category Personals-Dating
```

Example Configuring Parameter Maps for Trend Micro URL Filtering

The following example shows a parameter map for global Trend Micro parameters and a parameter map for per-policy Trend Micro parameters:

```
parameter-map type trend-global global-param-map
 server trps1.trendmicro.com retrans 5 timeout 200
 cache-entry-lifetime 1
 cache-size maximum-memory 128000
parameter-map type urlfpolicy trend trend-param-map
 block-page message "group2 is blocked by trend"
 max-request 2147483647
 max-resp-pak 20000
 truncate hostname
```

Example Attaching a URL Filtering Policy

The following example configures an HTTP traffic class and an inspect type policy map that inspects all HTTP traffic, applies the URL filtering policy to that traffic, and ignores all other traffic. Finally, the inspect policy is attached as a service policy to the target zone pair.

```
class-map type inspect match-all http-class
 match protocol http
policy-map type inspect urlfilter trend-global-policy
 class type inspect http-class
  inspect global
  service-policy urlfilter trend-policy
 class class-default
  drop
zone-pair security zp-in source zone-in destination zone-out
 service-policy type inspect trend-global-policy
```

Example Subscription-Based Content Filtering Sample Configuration

The following sample subscription-based content filtering configuration specifies two different URL filtering policies--one for group one and one for group two:

```
! port map to indicate FW that all 8080 connections are http connections
ip port-map http port 8080
! Trend global parameter-map to specify the TRPS server and cache-sizes
parameter-map type trend-global hello
 server trps1.trendmicro.com
 cache-size maximum-memory 300
! Trend Policy parameter map for group one.
!   If server is down, allow the HTTP connections
parameter-map type urlfpolicy trend trend-g1-param
 allow-mode on
 block-page message "You are prohibited from accessing this web page"
! Trend Policy parameter map for group two.
!   If the server is down block the HTTP connections
parameter-map type urlfpolicy trend trend-g2-params
 block-page message "Restricted access. Please contact your administrator"
! Trend class map for group one
!   Just match bad reputation sites
class-map type urlfilter trend trend-g1-c
 match url reputation ADWARE
 match url reputation DIALER
! Trend class map for group two
!   Match on bad reputation sites and on Gambling and Personals-Dating sites
class-map type urlfilter trend trend-g2-c
 match url reputation ADWARE
 match url reputation PHISHING
 match url category Gambling
 match url category Personals-Dating
! Local filtering class to permit certain domains
parameter-map type urlf-glob p-domains
 pattern "www.example.com"
 pattern "www.example1.com"
class-map type urlfilter p-domains
   match server-domain urlf-glob p-domains
! Local filtering class to deny certain domains
parameter-map type urlf-glob d-domains
 pattern "*.example2.com"
 pattern "www.example3.com"
class-map type urlfilter d-domains
   match server-domain urlf-glob d-domains
! Urlfilter Policy map for group one.
! Don't block any of the domains locally
policy-map type inspect urlfilter g1-pol
```

```
 parameter type urlfpolicy trend trend-g1-param
 class type urlfilter p-domains
  allow
 class type urlfilter d-domains
  reset
 class type urlfilter trend trend-g1-c
  reset
! Url filter policy map for group two
!   Block the deny domains locally
policy-map type inspect urlfilter g2-pol
 parameter type urlfpolicy trend trend-g2-param
 class type urlfilter p-domains
  allow
 class type urlfilter d-domains
  log
  reset
 class type urlfilter trend trend-g2-c
  reset
! First level class to prevent content filtering for websites that are local to the enterprise
! The first deny line is to make the http connections going to the proxy to not match this
 class
ip access-list extended 101
 deny tcp any host 192.168.1.10 eq 8080
 permit tcp any 192.168.0.0 0.0.255.255 eq 80 8080
 permit tcp any 10.0.0.0 0.255.255.255 eq 80 8080
class-map type inspect no-urlf-c
   match access-group 101
! First level class map to support url-filtering for group one
ip access-list extended 102
 permit tcp 192.168.1.0 0.0.0.255 any
class-map type inspect urlf-g1-c
 match protocol http
 match access-group 102
! First level class map to support url-filtering for group two
ip access-list extended 103
 permit tcp 192.168.2.0 0.0.0.255 any
class-map type inspect urlf-g1-c
 match protocol http
 match access-group 103
! First level class map to allow ICMP from protected network to outside
class-map type inspect icmp-c
   match protocol icmp
! First level policy map that brings everything together
! Always configure the class with most restrictions first
policy-map type inspect fw-pol
 class type inspect icmp
  inspect
 class type inspect no-urlf-c
  inspect
 class type inspect urlf-g2-c
  inspect
  service-policy urlfilter g2-pol
 class type inspect urlf-g1-c
  inspect
  service-policy urlfilter g1-pol
! Create targets to which the FW policy is applied
zone security z1
zone security z2
zone-pair security z1z2 source z1 destination z2
 service-policy type inspect fw-pol
! inside interface
interface FastEthernet 0/0
 ip address 10.1.1.1 255.255.0.0
 zone-member security z1
!outside interface
interface FastEthernet 1/0
 ip address 209.165.200.225 255.255.255.224
 zone-member security z2
```

Example Configuring URL Filtering with a Websense Server

The following example configures URL filtering with a Websense server:

```
parameter-map type urlfpolicy websense websense-param-map
/* define vendor related info */
 server 192.168.3.1
 port 5000 retrans 3 timeout 200
/* define global info related with URL filtering */
 alert on
 allow-mode off
 urlf-server-log on
 max-request 2000
 max-resp-pak 200
 truncate hostname
 cache-size 256
 cache-entry-lifetime 2
 block-page "This page has been blocked."

/* define trusted-domain lists */
! Local filtering class to permit certain domains
parameter-map type urlf-glob p-domains
 pattern "www.example.com"
 pattern "www.example1.com"
class-map type urlfilter p-domains
 match server-domain urlf-glob p-domains
! Local filtering class to deny certain domains
parameter-map type urlf-glob d-domains
 pattern "*.example2.com"
 pattern "www.example3.com"
class-map type urlfilter d-domains
 match server-domain urlf-glob d-domains
class-map type urlfilter websense match-any websense-map
 match server-response any
policy-map type inspect urlfilter url-websense-policy
 parameter-map urlfpolicy websense websense-param-map
 class type urlfilter trusted-domain-lists
  allow
 class type urlfilter untrusted-domain-lists
  reset
 class type urlfilter block-url-keyword-lists
  reset
 class type urlfilter websense websense-map
  server-specified-action
/* define customer group */
access-list 101 permit ip 192.168.1.0 0.0.0.255 any
class-map type inspect match-all urlf-traffic
 match protocol http
 match access-list 101
policy-map type inspect urlfilter-policy
 class type inspect urlf-traffic
  inspect
  service-policy urlfilter url-websense-policy
```

Example Configuring URL Filtering with a SmartFilter Server

The following example configures URL filtering with a SmartFilter server:

```
parameter-map type urlfpolicy n2h2 n2h2-param-map
/* define vendor related info */
 server 192.168.3.1
 port 5000 retrans 3 timeout 200
/* define global info related with URL filtering */
 alert on
 allow-mode off
```

```
 urlf-server-log on
 max-request 2000
 max-resp-pak 200
 truncate hostname
 cache-size 256
 cache-entry-lifetime 2
 block-page “This page has been blocked.”
/* define trusted-domain lists */
! Local filtering class to permit certain domains
parameter-map type urlf-glob p-domains
 pattern "www.example.com"
 pattern "www.example1.com"
class-map type urlfilter p-domains
 match server-domain urlf-glob p-domains
! Local filtering class to deny certain domains
parameter-map type urlf-glob d-domains
 pattern "*.example2.com"
 pattern "www.example3.com"
class-map type urlfilter d-domains
 match server-domain urlf-glob d-domains
class-map type urlfilter websense match-any n2h2-map
 match server-response any
policy-map type inspect urlfilter url-n2h2-policy
 parameter-map urlfpolicy n2h2 n2h2-param-map
  class type urlfilter trusted-domain-lists
   allow
  class type urlfilter untrusted-domain-lists
  reset
  class type urlfilter block-url-keyword-lists
   reset
  class type urlfilter n2h2 n2h2-map
   server-specified-action
/* define customer group */
access-list 101 permit ip 192.168.1.0 0.0.0.255 any
class-map type inspect match-all urlf-traffic
 match protocol http
 match access-list 101
policy-map type inspect urlfilter-policy
 class type inspect urlf-traffic
  inspect
  service-policy urlfilter url-n2h2-policy
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
The Cisco IOS firewall solution	Cisco IOS Firewall Overview

Standards

Standard	Title
None	--

MIBs

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

RFCs

RFC	Title
RFC 1945	*Hypertext Transfer Protocol--HTTP/1.0*
RFC 2616	*Hypertext Transfer Protocol--HTTP/1.1*

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 Subscription-Based Cisco IOS Content Filtering

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

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

Table 24: Feature Information for Subscription-Based Cisco IOS Content Filtering

Feature Name	Releases	Feature Information
Cisco IOS Content Filtering	12.4(15)XZ 12.4(20)T	This feature interacts with the Trend Micro URL filtering service so that HTTP requests can be allowed, blocked, or logged, based on a content filtering policy. The content filtering policy specifies how to handle items such as categories, reputations (or security ratings), trusted domains, untrusted domains, and keywords. The following commands were introduced or modified: **class-map type urlfilter**, **class type urlfilter**, **clear zone-pair urlfilter cache**, **debug cce dp named-db urlfilter**, **debug ip trm**, **debug ip urlfilter**, **match server-domain urlf-glob**, **match server-response anymatch url category**, **match url reputation**, **match url- keyword urlf-glob**, **parameter-map type trend-global**, **parameter-map type urlf-glob**, **parameter-map type urlfpolicy**, **policy-map type inspect urlfilter**, **show class-map type urlfilter**, **show ip trm config**, **show ip trm subscription status**, **show parameter-map type trend-global**, **show parameter-map type urlf-glob**, **show parameter-map type urlfpolicy**, **show policy-map type inspect urlfilter**, **show policy-map type inspect zone-pair**, **show policy-map type inspect zone-pair urlfilter**, **trm register**.

CHAPTER 13

Cisco IOS Firewall Support for Skinny Local Traffic and CME

The Cisco IOS Firewall Support for Skinny Local Traffic and CME feature enhances the Context-Based Access Control (CBAC) functionality to support Skinny traffic that is either generated by or destined to the router. When Cisco Call Manager Express (CME) is enabled on the Cisco IOS firewall router, the CME manages both VoIP and analog phones using Skinny Client Control Protocol (SCCP) over either an intranet or the Internet with flow-around and flow-through modes of CME.

In addition,the Firewall Support of Skinny Client Control Protocol feature extends the support of SCCP to accommodate video channels.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Cisco IOS Firewall Support for Skinny Local Traffic and CME

The Skinny inspection module is part of the inspection subsystem; thus, your router must be running an image that has firewall support.

Restrictions for Cisco IOS Firewall Support for Skinny Local Traffic and CME

This feature has the following restrictions:

- Skinny inspection will inspect only the SCCP sessions that have been established after the firewall is configured with Skinny inspection. That is, any SCCP sessions that were established through the firewall before the Skinny inspection was configured will not be inspected.
- This feature does not support Music on Hold (MOH) when a device other than the Call Manager (CM) is the music server. (This feature does support MOH when the CM is the music server.)
- This feature does not address either the multicast functionality of SCCP or the functionality of multiple active calls on a single Skinny client.

This feature does not support the following Skinny and firewall configurations.

The CM and the Skinny client cannot be on three different networks that are separated at the firewall. The firewall implementation does not inspect sessions that have devices residing on more than two distinct networks that are segregated at the firewall. That is, if more than two interfaces at the firewall, session inspection is not supported.

Information About Cisco IOS Firewall Support for Skinny Local Traffic and CME

Skinny Inspection Overview

Skinny inspection enables voice communication between two Skinny clients by using the Cisco CallManager. The Cisco CallManager uses the TCP port 200 to provide services to Skinny clients. A Skinny client connects to the primary Cisco CallManager by establishing a TCP connection and if available, connects to a secondary Cisco CallManager. After the TCP connection is established, the Skinny client registers with the primary Cisco CallManager, which will be used as the controlling Cisco CallManager until it reboots or a keepalive failure occurs. Thus, the TCP connection between the Skinny client and the Cisco CallManager exists forever and is used to establish calls coming to or from the client. If a TCP connection fails, the secondary Cisco CallManager is used. All data channels established with the initial Cisco CallManager remain active and will be closed after the call ends.

The Skinny protocol inspects the locally generated or terminated Skinny control channels and opens or closes pinholes for media channels that originate from or are destined to the firewall. Pinholes are ports that are opened through a firewall to allow an application controlled access to a protected network. The Skinny traffic that passes through and locally generated or terminated Skinny traffic is treated in the same way at the firewall.

The table below lists the set of messages that are necessary for the data sessions to open and close. Skinny inspection will examine the data sessions that are deemed for opening and closing the access list pinholes.

Table 25: Skinny Data Session Messages

Skinny Inspection Message	Description
StationCloseReceiveChannelMessage	Sent by Cisco CallManager instructing the Skinny client (on the basis of the information in this message) to close the receiving channel.
StationOpenReceiveChannelAckMessage	Contains the IP address and port information of the Skinny client sending this message. This message also contains the status of whether or not the client is willing to receive voice traffic.
StationStartMediaTransmissionMessage	Contains the IP address and port information of the remote Skinny client.
StationStopMediaTransmissionMessage	Sent by the Cisco CallManager instructing the Skinny client (on the basis of the information in this message) to stop transmitting voice traffic.
StationStopSessionTransmissionMessage	Sent by the Cisco CallManager instructing the Skinny client (on the basis of the information in this message) to end the specified session.
StationOpenMultiMediaReceiveChannelAckMessage	Contains the IP address and port information of the Skinny client sending this message. It also contains the status of whether the client is willing to receive video and data channels.
StationCloseMultiMediaReceiveChannel	Sent by the Cisco Unified Communications Manager to the Skinny endpoint to request the closing of the receiving video or data channel.
StationStartMultiMediaTransmitMessage	Sent by the Cisco Unified Communications Manager to the Skinny endpoint whenever Cisco Unified Communications Manager receives an OpenLogicalChannelAck message for the video or data channel.
StationStopMultiMediaTransmission	Sent to Skinny endpoints to request the stopping of the transmission of video or data channel.

Pregenerated Session Handling

When two phones register with the CME running on Cisco IOS firewall, two control channels terminated on the CME box. These two control channels are TCP connections and are inspected by the Firewall Skinny module. When pinholes are opened for the media traffic, a total of four pre-gen sessions are created, two for each control session.

With the flow-through mode of operation of CME, the four pregenerated sessions are converted to two active sessions. The same number of active sessions is retained because there are two media sessions, one from each phone terminating on CME.

With the flow-around mode of operation of CME, the CME is bypassed as there is a direct connection between the two phones. In this mode, there are two possible scenarios:

- When both phones are on the same side of the CME, there is no exchange of media packets between the two phones. However, exchange of media packets is possible with pass-through traffic. In this case, the pre-gen sessions will timeout because the media traffic will not reach the router itself.
- When both phones are located on either side of the CME, the media traffic goes through the CME box. The four pre-gen sessions that are created are converted to one active session. Instead of creating four pre-gen sessions, only two pre-gen sessions are created. These two pre-gen sessions are converted to one active session when you see the media traffic.

NAT with CME and the Cisco IOS Firewall

In typical deployments, both Cisco IOS firewall and Network Address Translator (NAT) will be running on the same router. When CME is also running, typically in the case of an Integrated Services Router (ISR), some complexities and limitations exist.

- If two Skinny phones are registered to CME that is on the Cisco IOS firewall with NAT. When Phone 1 attempts to communicate with Phone 2, the IP and port (mostly private IP) of Phone 1 will be exchanged with Phone 2 over the already established TCP connection.
- If NAT is configured on the outside interface to translate all the private addresses to the router's global address. Some private addresses are exchanged over a TCP connection between the router and the remote phone. If NAT is able to translate the addresses in such flows where one endpoint is the router itself, then NAT and CME running on the same box will not cause any problems. If not, the following scenarios are possible:
- In flow-through mode of operation, the voice data channels, Real-time Transport Protocol (RTP) stream over User Datagram Protocol (UDP), from Phone 1 and Phone 2, both terminate on CME. So, there will be one RTP over UDP connection from Phone 1 to the CME and a second from Phone 2 to the CME. The CME relays the voice data over the two channels. In this case, there should not be any problem with NAT running on the CME box, as the connection is terminated on the router from Phone 2 and the address used for that connection is the global address of the router.
- In flow-around mode of operation, there is a direct connection (RTP over UDP) between Phone1 and Phone 2 for carrying voice data traffic. If NAT does not translate the private IP of Phone 1, then the voice data channel will not be established successfully because the private IP of Phone 1 is shared in the control channel. In such a scenario, the running of CME with NAT breaks down.

New Registry for Locally Generated Traffic

A new registry is created in the Skinny local media traffic path. This path differs from the regular switching path code, where all the controlling and pass-through media traffic is inspected. The Skinny module sends the locally generated traffic using the "fastsend" application program interface (API) which does not put the packet in the regular switching path, but sends it directly (to Layer 2 drivers). This new registry resets the timeouts for the media channels and also reports the number of Skinny media sessions that are established such as the output of **show** commands.

Note The above API is used to update the Firewall sessions when the media channel is active. Firewall will not attempt to protect the CME box based on the nonexistence of pregen. Therefore, the firewall will not drop media packets for which there is no pre-gen/active session. The MTP module in CME protects itself by dropping the packets that do not match the source IP and source port numbers.

How to Configure Cisco IOS Firewall Support for Skinny Local Traffic and CME

Creating a ZonePair Between a Zone and the Self Zone

To inspect the traffic that is destined to the router or the traffic originating from the router, you need to create a zonepair between a zone (containing the incoming/outgoing interface) and the self zone.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **alert** {**on** | **off**}
5. **audit-trail** {**on** | **off**}
6. **class-map type inspect** protocol-name [**match-any**| **match-all**] class-map-name
7. **policy-map type inspect** *policy-map-name*
8. **class type inspect** *class-map-name*
9. **zone security** *name*
10. **zone security** *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*
14. **interface** *type number*
15. **zone-member security** *zone-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 inspect** *parameter-map-name* **Example:** `Router(config)# parameter-map type inspect insp-pmap`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action. • Enters parameter-map type inspect configuration mode.
Step 4	**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 5	**audit-trail** {**on** | **off**} **Example:** `Router(config-profile)# audit-trail on`	(Optional) Turns audit trail messages on or off.
Step 6	**class-map type inspect** protocol-name [**match-any**| **match-all**] class-map-name **Example:** `Router(config-profile)# class-map type inspect skinnycmap match-any protocol skinny`	Creates a class map for the Skinny protocol so that you can enter match criteria. • Enters class-map configuration mode.
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Router(config-profile)# policy-map type inspect skinnypmap`	Creates a policy map so that you can enter match criteria. • Enters policy map configuration mode.

	Command or Action	Purpose
Step 8	**class type inspect** *class-map-name* **Example:** `Router(config-profile)#` **class type inspect** *skinnycmap*	Specifies the name of the class on which an action is to be performed. • The value of the *class-map-name* argument must match the appropriate class name specified via the **class-map type inspect** command.
Step 9	**zone security** *name* **Example:** `Router(config-profile)# zone security z1`	Creates a zone for phone 1. • Enters global configuration mode.
Step 10	**zone security** *name* **Example:** `Router(config-profile)# zone security z2`	Creates a zone for phone 2.
Step 11	exit **Example:** `Router(config-profile)#exit`	Exits profile 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 z1-self source z1 destination self`	Creates a zone-pair. • 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 skinnypmap`	Attaches a firewall policy map to the destination zone-pair. • If a policy is not configured between a pair of zones, traffic is dropped by default. • Enters global configuration mode.
Step 14	**interface** *type number* **Example:** `Router(config)# interface FastEthernet4/1`	Specifies the type of interface to be configured and the port, connector, or interface card number.

	Command or Action	Purpose
Step 15	**zone-member security** *zone-name* **Example:** `Router(config-sec-zone-pair)#` **zone-member security** *z1*	Specifies the name of the security zone to which an interface is attached.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Firewall support of SCCP	"Firewall Support of Skinny Client Control Protocol (SCCP)" chapter in th e *Cisco IOS Security Configuration Guide*
Firewall commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
None	--

MIBs

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

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 Cisco IOS Firewall Support for Skinny Local Traffic and CME

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

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

Table 26: Feature Information for Cisco IOS Firewall Support for Skinny Local Traffic and CME

Feature Name	Releases	Feature Information
IOS Firewall Support for Skinny Local Traffic and CME	12.4(20)T	The Cisco IOS Firewall Support for Skinny Local Traffic and CME feature enhances the Context-Based Access Control (CBAC) functionality to support 'router generated/destined to router' Skinny traffic. When CME is enabled on the IOS firewall router, it manages both VoIP and analog phones using Skinny Client Control Protocol (SCCP) over intranet or internet with flow-around and flow-through modes of CME. The following commands were introduced or modified: **class-map type inspect, class type inspect, interface, parameter-map type inspect, policy-map type inspect, service-policy type inspect, zone-member security, zone-pair security.**
IOS Zone-Based Firewall SCCP Video Support	15.1(2)T	The IOS Zone-Based Firewall SCCP Video Support (SCCP) feature extends support to accommodate video channels.

CHAPTER 14

User-Based Firewall Support

Firewalls traditionally apply rules based on source and destination IP addresses. In the new, highly dynamic mobile world, IP addresses of end systems constantly change. Therefore it becomes increasingly difficult to have a particular user group function assigned to a particular block of IP addresses. It is also difficult to apply firewall policies for a user group that is the source of the traffic. This feature allows source IP addresses to be associated with user groups. Network administrators can apply firewall policies based on user-groups, and the infrastructure can seamlessly apply these security policies.

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 User-Based Firewall Support section.

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for User-Based Firewall Support

Hardware Requirements

- Access Control Server
- Cisco Network Access Device, which can be any of the following:
  - Cisco 7200 router
  - Cisco 1800 router
  - Cisco 2800 router
  - Cisco 3800 router

Software Requirements

- Cisco IOS Release 12.4(20)T or a later release
- An Ingress Security feature that uses the Identity Policy infrastructure for policy application

Restrictions for User-Based Firewall Support

User-group mapping is based on the IPv4 address of the end-host's source. The "user-group" match criterion is supported for inspect class-maps.

Authentication Proxy and IP Admission

Authentication Proxy and IP Admission is an input-only feature that should be configured on all the interfaces of the source zone. The Authentication Proxy and IP Admission feature is not virtual routing and forwarding (VRF)-aware; therefore, the user-group Zone Policy Firewall policies cannot be applied on a per VRF basis.

Information About User-Based Firewall Support

Feature Design of User-Based Firewall Support

The User-Based Firewall Support feature was designed to provide identity or user-group based security that provides differentiated access for different classes of users. Classification can be provided on the basis of user identity, device type (for example, IP phones), location (for example, building) and role (for example, engineer). Because of the dynamic nature of end-host access, where every user is different and the resource he or she accesses is different, it is important to associate end-user's identity, role, or location with security policies. This association prevents the need for administrators to constantly update policy filters, a cumbersome task. The end-user identity can be derived through a variety of different mechanisms. Once a user's identity is established, security policies will be aware of the user's identity, not just the source address. Individual policies can be enforced allowing for greater control.

Cisco IOS supports several features that offer dynamic, per-user authentication and authorization of network access connections. These features include 802.1X, IKE, Authentication Proxy, Network Admission Control (NAC), and so on. These features allow network administrators to enforce security policies on per-user basis. By integrating authentication features with Cisco Policy Language-based features such as Zone Based Firewall, quality of service (QoS), and so on, the combination can provide a transparent, reliable, ease to manage and deploy security solution to dynamically authenticate and enforce polices on a per user basis.

Cisco IOS User-Based Firewall Support leverages existing authentication and validation methods to associate each source IP address to a user-group. User-group association can be achieved using two methods. The first method (Tag and Template) uses locally defined policies to achieve the association, while the second method obtains the user-group information from the access control server (ACS) and requires no further configuration on the network access device (NAD).

The User-Based Firewall Support feature leverages the Tag and Template concept where the authenticating server returns a tag-name on validating the user credentials. This tag received on the authentication device is mapped to a template. The template is a control plane policy map that refers to an identity policy configured on the device. The identity policy contains the access policies that are to be applied for the corresponding tag-name. The identity policy defines one or more user-groups to which the source IP would be associated. This mapping provides administrators with flexibility to associate the end-host with multiple user-group memberships. The scope of the user-group defined in the identity policy is local to the device. Once the end-host's user-group membership has been established, other Cisco IOS policy language based features can enforce security policies on a per user-group basis.

Match Criterion

The match user-group criterion in the inspect type class map configuration can be used to enforce security policies on a per user-group basis. The match criterion filters the traffic stream based on the client's source IP address in the specified user-group, making it independent of the authentication method that established the group membership. The match criterion in the inspect type class map enables inspection for any ingress traffic and for any protocol, thereby enabling inspection for all traffic.

Firewall Support

Cisco IOS Firewall includes multiple security features. Cisco IOS Firewall stateful packet inspection provides true firewall capabilities to protect networks against unauthorized traffic and control legitimate business-critical data. Authentication proxy controls access to hosts or networks based on user credentials stored in an authentication, authorization, and accounting (AAA) server. Multi-VRF firewall offers firewall services on virtual routers with VRF, accommodating overlapping address space to provide multiple isolated private route spaces with a full range of security services. Transparent firewall adds stateful inspection without time-consuming, disruptive IP addressing modifications. Application inspection controls application activity to provide granular policy enforcement of application usage, protecting legitimate application protocols from rogue applications and malicious activity. For more information on firewall support see the Cisco IOS Firewall Design Guide.

Authentication Proxy

The Cisco IOS Firewall Authentication Proxy feature provides dynamic, per-user authentication and authorization, authenticating users against industry standard TACACS+ and RADIUS authentication protocols. Authenticating and authorizing connections by users provides more robust protection against network attacks. See the Authentication Proxy document for more information about this feature.

Zone-Based Policy Firewall

Cisco IOS Zone-Based Policy Firewall can be used to deploy security policies by assigning interfaces to different zones and configuring a policy to inspect the traffic moving between these zones. The policy specifies a set of actions to be applied on the defined traffic class. For more information see the document Zone-Based Firewall.

Tag and Template

The Tag and Template feature allows network administrators to define enforcement policies on a local device and have a RADIUS server specify the policy selector to be enforced. This feature can be applied to a NAC architecture. See the Tag and Template feature guide for more information about this feature.

Network Admission Control

In a typical Network Admission Control deployment, an ACS or a RADIUS server is used for validating the user posture information and for applying the policies on the NAD. A centralized ACS can be used to support multiple NADs. This solution has inherent problems associated with it, namely:

- Version control of policies. Typically, a specific NAD that is running a Cisco IOS image may support some access control lists (ACLs), and another NAD may support a different version. Managing different versions can be a problem.
- Users connect on different interfaces to the NAD, and on the basis of the interface type, the policies that can be applied to the user can change, and the NAD can determine the policies to be applied. In the current architecture, the ACS sends the same set of policies to all the NADs when a profile is matched, which does not give enough control to the administrator to configure the polices on the basis of the NAD configuration.

Configuring the Tag and Template feature allows the ACS to map users to specific groups and associate a tag with them. For example, the Usergroup1 user group may have a tag with the name usergroup1. When the NAD queries the ACS for the policies, the ACS can return the tag that is associated with the user group. When this tag is received at the NAD, the NAD can map the tag to a specific template that can have a set of policies that are associated with the user group. This mapping provides administrators with the flexibility to configure the template on a NAD basis, and the policies can change from NAD to NAD even though the tag is the same.

In summary, a template must be configured on the NAD, and the template must be associated with a tag. When the ACS sends the policies back to the NAD, the template that matches the tag that was received from the ACS is used.

Access Control List Overview

Cisco provides basic traffic filtering capabilities with access control lists (also referred to as access lists). Access lists can be configured for all routed network protocols (IP, AppleTalk, and so on) to filter the packets of those protocols as the packets pass through a router. You can configure access lists at your router to control access to a network. Access lists can prevent certain traffic from entering or exiting a network.

How to Configure User-Based Firewall Support

Configuring Access Control Lists

To configure ACLs, perform the steps in this section. Policy specific ACLs are defined under the identity policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. **permit** *protocol* **any host** *ip-address*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended auth_proxy_acl`	Defines an IP access list and enters extended named access list configuration mode.

	Command or Action	Purpose
Step 4	**permit** *protocol* **any** **host** *ip-address* **Example:** `Router(config-ext-nacl)# permit tcp any host 192.168.104.136`	Sets the permission for an access list using TCP.
Step 5	**end** **Example:** `Router(config-ext-nacl)# end`	Exits extended named access list configuration mode.

Configuring the Identity Policy for Tag and Template

To configure the identity policy for Tag and Template, perform the steps in this section. Usergroup support is achieved by configuring the usergroup that is to be associated with the IP address on the NAD itself using a locally defined identity policy. A tag is received from the ACS that matches a template (identity policy) on the NAD. The user-group associated with the IP address is obtained from the NAD.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **identity policy** *policy-name*
4. **user-group** *group-name*
5. **access-group** *group-name*
6. **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.

	Command or Action	Purpose
Step 3	**identity policy** *policy-name* **Example:** `Router(config)# identity policy auth_proxy_ip`	Creates an identity policy and enters identity policy configuration mode.
Step 4	**user-group** *group-name* **Example:** `Router(config-identity-policy)# user-group auth_proxy_ug`	Establishes a user-group.
Step 5	**access-group** *group-name* **Example:** `Router(config-identity-policy)# access-group auth_proxy_acl`	Specifies the access-group to be applied to the identity policy.
Step 6	**end** **Example:** `Router(config-identity-policy)# end`	Exits identity policy configuration mode.

Configuring Control Type Tag Class-Maps or Policy-Maps for Tag and Template

To configure control type tag class-maps or policy-maps for Tag and Template, perform the steps in this section. Tag names are received from the AAA server as authorization data and are matched with their respective class-maps. The security policies that are associated with the identity policies are applied to the host. In this way host IP addresses gain membership of user-groups.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type control tag** *policy-map-name*
4. **class type control tag** *control-class-name*
5. **identity policy** *policy-name*
6. **exit**
7. **configure terminal**
8. **class-map type control tag match-all** *class-map-name*
9. **match tag** *tag-name*
10. **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	**policy-map type control tag** *policy-map-name* **Example:** `Router(config)# policy-map type control tag all_tag_cm_pm`	Creates a control policy map and enters policy-map configuration mode.
Step 4	**class type control tag** *control-class-name* **Example:** `Router(config-pmap)# class type control tag auth_proxy_tag_cm`	Creates a control class and enters policy-map-class configuration mode.
Step 5	**identity policy** *policy-name* **Example:** `Router(config-pmap-c)# identity policy auth_proxy_ip`	Creates an identity policy.
Step 6	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map-class configuration mode.
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**class-map type control tag match-all** *class-map-name* **Example:** `Router(config)# class-map type control tag match-all auth_proxy_tag_cm`	Creates a control class map and enters class-map configuration mode.

	Command or Action	Purpose
Step 9	**match tag** *tag-name* **Example:** `Router(config-cmap)# match tag auth_proxy_tag`	Specifies the tag to be matched for a tag type of class map.
Step 10	**end** **Example:** `Router(config-cmap)# end`	Exits class-map configuration mode.

Configuring Supplicant-Group Attribute on the ACS

The supplicant group attribute needs to be configured as a Cisco attribute value (AV) Pair on the ACS for user-based firewall support. To configure the supplicant-group attribute on the ACS, perform the steps in this section. The supplicant-group attribute is defined in the RADIUS and Lightweight Directory Access Protocol (LDAP) authorization group attributes from where all authorization data pertaining to the client resides. The user-group information is obtained from the ACS and no further user-group specific configuration is required on the NAD.

Cisco:Avpair=supplicant-group=eng

Defines the supplicant-group attribute.

Configuring Firewall Class-Maps and Policy-Maps

Perform the following task to configure firewall class-maps and policy-maps. User-groups are configured and attached to policy-maps by using the **inspect** command with each class-map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-all** *class-map-name*
4. **match protocol** *protocol-name*
5. **match user-group** *group-name*
6. **exit**
7. **configure terminal**
8. **policy-map type inspect** *policy-map-name*
9. **class type inspect** *class-map-name*
10. **inspect**
11. **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	**class-map type inspect match-all** *class-map-name* **Example:** `Router(config)# class-map type inspect match-all auth_proxy_ins_cm`	Creates an inspect type class map and enters class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol telnet`	Configures the match criterion for the class map on the basis of the specified protocol.
Step 5	**match user-group** *group-name* **Example:** `Router(config-cmap)# match user-group auth_proxy_ug`	Configures the match criterion for the class map on the basis of the specified user-group.
Step 6	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect all_ins_cm_pm`	Creates an inspect type policy map and enters policy-map configuration mode.

	Command or Action	Purpose
Step 9	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect auth_proxy_ins_cm`	Specifies the traffic (class) on which an action is to be performed.
Step 10	**inspect** **Example:** `Router(config-pmap)# inspect`	Enables Cisco IOS stateful packet inspection.
Step 11	**end** **Example:** `Router(config-pmap)# end`	Exits policy-map configuration mode.

Configuring Firewall Zone Security and Zone-Pair

To configure firewall zone security and zone -pair, perform the steps in this section. Security zones are configured for untrustworthy (outside) and trustworthy (inside) networks or interfaces. Zone-pairs are configured where the source zone is untrustworthy and the destination zone is trustworthy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **end**
5. **configure terminal**
6. **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*
7. **service-policy type inspect** *policy-map-name*
8. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Router> enable`	• 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 out_sec_zone`	Creates a security zone, and enters security zone configuration mode.
Step 4	**end** **Example:** `Router(config-sec-zone)# end`	Exits security zone configuration mode.
Step 5	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 6	**zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name* **Example:** `Router(config)# zone-pair security out_in source out_sec_zone destination in_sec_zone`	Creates a zone-pair and enters security zone-pair configuration mode.
Step 7	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect all_ins_cm_pm`	Attaches a firewall policy map to the zone-pair.
Step 8	**end** **Example:** `Router(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode.

Configuring ACLs for Authentication Proxy

To configure ACLs for authentication proxy, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. **permit** *protocol* **any** *source-ip-address destination-ip-address*
5. **permit** *protocol* **any host** *destination-ip-address*
6. **permit** *protocol* **any any eq bootps**
7. **permit** *protocol* **any any eq domain**
8. **end**
9. **configure terminal**
10. **ip access-list extended** *access-list-name*
11. **permit** *protocol* **any host** *destination-ip-address*
12. **permit** *protocol* **any host** *destination-ip-address* **eq domain**
13. **permit** *protocol* **any host** *destination-ip-address* **eq www**
14. **permit** *protocol* **any host** *destination-ip-address* **eq** *port*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended 102`	Defines an IP access list and enters extended named access list configuration mode.
Step 4	**permit** *protocol* **any** *source-ip-address destination-ip-address*	Sets the permission for an access list using IP.

	Command or Action	Purpose
	Example: `Router(config-ext-nacl)# permit ip any 192.168.100.0 10.0.0.255`	
Step 5	**permit** *protocol* **any host** *destination-ip-address* **Example:** `Router(config-ext-nacl)# permit ip any host 192.168.104.136`	Sets the permission for an access list using IP.
Step 6	**permit** *protocol* **any any eq bootps** **Example:** `Router(config-ext-nacl)# permit ip any any eq bootps`	Sets the permission for an access list using IP.
Step 7	**permit** *protocol* **any any eq domain** **Example:** `Router(config-ext-nacl)# permit ip any any eq domain`	Sets the permission for an access list using IP.
Step 8	**end** **Example:** `Router(config-ext-nacl)# end`	Exits extended named access list configuration mode.
Step 9	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 10	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended 103`	Defines an IP access list and enters extended named access list configuration mode.
Step 11	**permit** *protocol* **any host** *destination-ip-address* **Example:** `Router(config-ext-nacl)# permit ip any host 192.168.104.136`	Sets the permission for an access list using IP.

	Command or Action	Purpose
Step 12	**permit** *protocol* **any** **host** *destination-ip-address* **eq domain** **Example:** `Router(config-ext-nacl)# permit udp any host 192.168.104.136 eq domain`	Sets the permission for an access list using user datagram protocol (UDP).
Step 13	**permit** *protocol* **any** **host** *destination-ip-address* **eq www** **Example:** `Router(config-ext-nacl)# permit tcp any host 192.168.104.136 eq www`	Sets the permission for an access list using TCP.
Step 14	**permit** *protocol* **any** **host** *destination-ip-address* **eq** *port* **Example:** `Router(config-ext-nacl)# permit udp any host 192.168.104.136 eq 443`	Sets the permission for an access list using UDP.
Step 15	**end** **Example:** `Router(config-ext-nacl)# end`	Exits extended named access list configuration mode.

Configuring Authentication Proxy

To configure authentication proxy default IP admissions, perform the steps in this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission auth-proxy-banner http c** *Auth-Proxy-Banner-Text* **c**
4. **ip admission watch-list expiry-time** *expiry-minutes*
5. **ip admission max-login-attempts** *attempt-number*
6. **ip admission inactivity-timer** *timeout-minutes*
7. **ip admission absolute-timer** *timeout-minutes*
8. **ip admission init-state-timer** *timeout-minutes*
9. **ip admission auth-proxy-audit**
10. **ip admission watch-list enable**
11. **ip admission ratelimit** *limit*
12. **ip admission name** *admission-name* **proxy http list** *acl*
13. **ip admission name** *admission-name* **proxy telnet list** *acl*
14. **ip admission name** *admission-name* **proxy http list** *acl* **service-policy type tag** *service-policy-name*
15. **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 admission auth-proxy-banner http c** *Auth-Proxy-Banner-Text* **c** **Example:** `Router(config)# ip admission auth-proxy-banner http c Auth-Proxy-Banner-Text c`	Creates a network admission control rule with an authentication proxy banner to be applied to the interface.
Step 4	**ip admission watch-list expiry-time** *expiry-minutes* **Example:** `Router(config)# ip admission watch-list expiry-time 50`	Creates a network admission control rule with a watch-list to be applied to the interface.

	Command or Action	Purpose
Step 5	**ip admission max-login-attempts** *attempt-number* **Example:** `Router(config)# ip admission max-login-attempts 10`	Creates a network admission control rule with a specified maximum login attempts per user number to be applied to the interface.
Step 6	**ip admission inactivity-timer** *timeout-minutes* **Example:** `Router(config)# ip admission inactivity-timer 205`	Creates a network admission control rule with a specified inactivity timeout to be applied to the interface.
Step 7	**ip admission absolute-timer** *timeout-minutes* **Example:** `Router(config)# ip admission absolute-timer 305`	Creates a network admission control rule with a specified absolute timeout to be applied to the interface.
Step 8	**ip admission init-state-timer** *timeout-minutes* **Example:** `Router(config)# ip admission init-state-timer 15`	Creates a network admission control rule with a specified init-state timeout to be applied to the interface.
Step 9	**ip admission auth-proxy-audit** **Example:** `Router(config)# ip admission auth-proxy-audit`	Creates a network admission control rule with authentication proxy auditing to be applied to the interface.
Step 10	**ip admission watch-list enable** **Example:** `Router(config)# ip admission watch-list enable`	Creates a network admission control rule with a watch-list to be applied to the interface.
Step 11	**ip admission ratelimit** *limit* **Example:** `Router(config)# ip admission ratelimit 100`	Creates a network admission control rule with a specified session rate limit to be applied to the interface.
Step 12	**ip admission name** *admission-name* **proxy http list** *acl* **Example:** `Router(config)# ip admission name auth_rule proxy http list 103`	Creates an IP network admission control rule. • Telnet, HTTP, or both can be configured.

	Command or Action	Purpose
Step 13	**ip admission name** *admission-name* **proxy telnet list** *acl* **Example:** `Router(config)# ip admission name auth_rule proxy telnet list 103`	Creates an IP network admission control rule. • Telnet, HTTP, or both can be configured.
Step 14	**ip admission name** *admission-name* **proxy http list** *acl* **service-policy type tag** *service-policy-name* **Example:** `Router(config)# ip admission name auth_rule proxy http list 103 service-policy type tag all_tag_cm_pm`	(Optional) Creates an IP network admission control rule. • Configures a control plane service policy when the Tag & Template method of user-group association is used. • Control plane tag service policy that is configured using the **policy-map type control tag** *policy name* command, keyword, and argument. This policy map is used to apply the actions on the host when a tag is received.
Step 15	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring AAA and RADIUS

To configure AAA and RADIUS servers, perform the steps in this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login default group radius**
5. **aaa authentication login** *list-name* **none**
6. **aaa authentication eou default enable group radius**
7. **aaa authorization network default group radius local**
8. **aaa authorization** *list-name* **default group radius**
9. **aaa accounting auth-proxy default start-stop group** *group-name*
10. **aaa accounting system default start-stop group** *group-name*
11. **aaa session-id common**
12. **radius-server attribute 6 on-for-login-auth**
13. **radius-server attribute 8 include-in-access-req**
14. **radius-server attribute 25 access-request include**
15. **radius-server configure-nas**
16. **radius-server host** *ip-address* **auth-port** *port-number* **acct-port** *port-number* **key** *string*
17. **radius-server host** *ip-address* **auth-port** *port-number* **acct-port** *port-number* **key** *string*
18. **radius-server source-ports extended**
19. **radius-server vsa send authentication**
20. **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	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.

	Command or Action	Purpose
Step 4	**aaa authentication login default group radius** **Example:** `Router(config)# aaa authentication login default group radius`	Sets AAA authentication at login using the group radius method.
Step 5	**aaa authentication login** *list-name* **none** **Example:** `Router(config)# aaa authentication login noAAA none`	Sets AAA authentication at login and ensures that the authentication succeeds even if all methods of authentication return an error.
Step 6	**aaa authentication eou default enable group radius** **Example:** `Router(config)# aaa authentication eou default enable group radius`	Sets authentication lists for Extensible Authentication Protocol over User Datagram Protocol (EAPoUDP).
Step 7	**aaa authorization network default group radius local** **Example:** `Router(config)# aaa authorization network default group radius local`	Sets parameters that restrict user access to a network using the group radius and local methods. • The group radius method uses the list of all RADIUS servers for authentication. • The local method uses the local database for authorization.
Step 8	**aaa authorization** *list-name* **default group radius** **Example:** `Router(config)# aaa authorization auth-proxy default group radius`	Sets parameters that restrict user access to a network using the group radius method.
Step 9	**aaa accounting auth-proxy default start-stop group** *group-name* **Example:** `Router(config)# aaa accounting auth-proxy default start-stop group radius`	Creates a method list to provide information about all authenticated-proxy user events. • Sends a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process.
Step 10	**aaa accounting system default start-stop group** *group-name* **Example:** `Router(config)# aaa accounting system default start-stop group radius`	Creates a method list to provide accounting for all system-level events not associated with users. • Sends a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process.

	Command or Action	Purpose
Step 11	**aaa session-id common** **Example:** `Router(config)# aaa session-id common`	Specifies that the same ID will be assigned for each AAA accounting service type within a call.
Step 12	**radius-server attribute 6 on-for-login-auth** **Example:** `Router(config)# radius-server attribute 6 on-for-login-auth`	Sends the Service-Type attribute in the authentication packets.
Step 13	**radius-server attribute 8 include-in-access-req** **Example:** `Router(config)# radius-server attribute 8 include-in-access-req`	Sends the IP address of a user to the RADIUS server in the access request.
Step 14	**radius-server attribute 25 access-request include** **Example:** `Router(config)# radius-server attribute 25 access-request include`	Sends an arbitrary value that the network access server includes in all accounting packets for the user if supplied by the RADIUS server.
Step 15	**radius-server configure-nas** **Example:** `Router(config)# radius-server configure-nas`	Configures the Cisco router or access server to query the vendor-proprietary RADIUS server for the static routes and IP pool definitions used throughout its domain when the device starts up.
Step 16	**radius-server host** *ip-address* **auth-port** *port-number* **acct-port** *port-number* **key** *string* **Example:** `Router(config)# radius-server host 192.168.104.131 auth-port 1645 acct-port 1646 key string1`	Specifies a RADIUS server host. • Specifies the UDP destination port for authentication requests. • Specifies the UDP destination port for accounting requests.
Step 17	**radius-server host** *ip-address* **auth-port** *port-number* **acct-port** *port-number* **key** *string* **Example:** `Router(config)# radius-server host 192.168.104.132 auth-port 1645 acct-port 1646 key string2`	Specifies a RADIUS server host. • Specifies the UDP destination port for authentication requests. • Specifies the UDP destination port for accounting requests.

	Command or Action	Purpose
Step 18	**radius-server source-ports extended** **Example:** `Router(config)# radius-server source-ports extended`	Enables 200 ports in the range from 21645 to 21844 to be used as the source ports for sending out RADIUS requests. • Ports 1645 and 1646 are used as the source ports for RADIUS requests.
Step 19	**radius-server vsa send authentication** **Example:** `Router(config)# radius-server vsa send authentication`	Configures the network access server (NAS) to recognize and use vendor-specific attributes (VSAs).
Step 20	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring AAA and LDAP

Perform this task to configure AAA and LDAP servers:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login default group ldap**
5. **aaa authentication login** list-name **none**
6. **aaa authorization network default group ldap local**
7. **aaa authorization** *list-name* **default group ldap**
8. **ldap attribute map** *map-name*
9. **map type** *ldap-attr-type aaa-attr-type*
10. **exit**
11. **ldap server** *name*
12. **ipv4** *ipv4-address*
13. **bind authenticate root-dn** *username* **password** [**0** *string* | **7** *string*] *string*
14. **base-dn** *string*
15. **attribute map** *map-name*
16. **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	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authentication login default group ldap** **Example:** `Router(config)# aaa authentication login default group ldap`	Sets AAA authentication at login using the group LDAP method.
Step 5	**aaa authentication login** list-name **none** **Example:** `Router(config)# aaa authentication login AAA none`	Sets AAA authentication at login and ensures that the authentication succeeds even if all methods of authentication return an error.
Step 6	**aaa authorization network default group ldap local** **Example:** `Router(config)# aaa authorization network default group ldap local`	Sets parameters that restrict user access to a network using the group LDAP and local methods. • The group LDAP method uses the list of all LDAP servers for authentication. • The local method uses the local database for authorization.
Step 7	**aaa authorization** *list-name* **default group ldap** **Example:** `Router(config)# aaa authorization auth-proxy default group ldap`	Sets parameters that restrict user access to a network using the group LDAP method.

	Command or Action	Purpose
Step 8	**ldap attribute map** *map-name* **Example:** `Router(config)# ldap attribute map map1`	Configures dynamic LDAP attribute map and enters attribute-map configuration mode.
Step 9	**map type** *ldap-attr-type aaa-attr-type* **Example:** `Router(config-attr-map)# map type supp-grp supplicant-group`	Defines an attribute map.
Step 10	**exit** **Example:** `Router(config-attr-map)# exit`	Exits the attribute-map configuration mode.
Step 11	**ldap server** *name* **Example:** `Router(config)# ldap server ldap_dir_1`	Specifies the LDAP server name and enters LDAP server configuration mode.
Step 12	**ipv4** *ipv4-address* **Example:** `Router(config-ldap-server)# ipv4 10.0.0.1`	Specifies the IP address of the LDAP server.
Step 13	**bind authenticate root-dn** *username* **password** [**0** *string* | **7** *string*] *string* **Example:** `Router(config-ldap-server)# bind authenticate root-dn "cn=administrator,cn=users, dc=cisco,dc=com password"`	Authenticates a client to a LDAP server.
Step 14	**base-dn** *string* **Example:** `Router(config-ldap-server)# base-dn dc=example,dc=sns,dc=com`	(Optional) Configures the base DN that you want to use to perform search operations in the LDAP directory tree.
Step 15	**attribute map** *map-name* **Example:** `Router(config-ldap-server)# attribute map map1`	Attaches the attribute map to a particular LDAP server.

	Command or Action	Purpose
Step 16	**exit** **Example:** `Router(config-ldap-server)# exit`	Exits LDAP server group configuration mode.

Troubleshooting Tips

The following commands can be used to troubleshoot User-Based Firewall Support:

- **clear ip admission cache**
- **debug user-group**
- **show debugging**
- **show epm session ip**
- **show ip access-lists**
- **show ip admission**
- **show logging**
- **show policy-map type inspect zone-pair**
- **show user-group**

Examples

show epm session ip

The following example shows sample output of the **show epm session**command when the **summary** keyword is used.

```
Router# show epm session ip summary
EPM Session Information
------------------------------
Total sessions seen so far: 8
Total Active sessions: 1
Session IP Address:
------------------------------
192.168.101.131
```

The following example shows sample output of the **show epm session**command when the *ip-address* argument is specified. The output below is displayed if a locally defined user-group association (Tag and Template method) is used.

```
Router# show epm session ip 192.168.101.131
Admission feature: Authproxy
Tag Received: eng_group_tag
Policy map used: all_tag_cm_pm
Class map matched: eng_tag_cm
```

The following example shows sample output of the **show epm session**command when the *ip-address* argument is specified. The output below is displayed if ACS defined (supplicant-group attribute configured on the ACS) user-group association is used.

```
Router# show epm session ip 192.168.101.131
Admission feature: Authproxy
AAA policies:
ACS ACL: xACSACLx-IP-TEST_ACL-47dfc392
Supplicant-Group: eng
Supplicant-Group: mgr
Proxy ACl: permit udp any any
Router#
```

show ip access-lists

The following example shows sample output of the **show ip access-lists**command.

```
Router# show ip access-lists
Extended IP access list 102
 permit icmp host 192.168.101.131 host 192.168.104.136 Auth-Proxy ACE downloaded from AAA
 permit udp host 192.168.101.131 host 192.168.104.136 Auth-Proxy ACE downloaded from AAA
 permit tcp host 192.168.101.131 host 192.168.104.136 Auth-Proxy ACE downloaded from AAA
10 permit ip any 192.168.100.0 10.0.0.255 (956 matches)
 20 permit ip any 192.168.101.0 10.0.0.255 (9 matches)
 30 permit ip any host 192.168.104.136 (20 matches)
 40 permit udp any any eq bootps
 50 permit udp any any eq domain
```

Extended IP access list 103

```
 10 permit ip any host 192.168.104.136 (3 matches)
 20 permit udp any host 192.168.104.136 eq domain
 30 permit tcp any host 192.168.104.136 eq www
 40 permit udp any host 192.168.104.136 eq 443
 50 permit tcp any host 192.168.104.136 eq 443
Extended IP access list vendor_group_acl
 10 permit ip any host 192.168.104.136
Extended IP access list auth_proxy_acl
 10 permit tcp any host 192.168.104.136
 20 permit udp any host 192.168.104.136
 30 permit icmp any host 192.168.104.136
Extended IP access list sales_group_acl
 10 permit ip any host 192.168.104.131
Extended IP access list eng_group_acl
 10 permit ip any host 192.168.100.132
Extended IP access list manager_group_acl
 10 permit ip any host 192.168.104.128
Router#
```

show ip admission

The following is sample output of the **show ip admission**command when the **configuration** keyword is used.

```
Router# show ip admission configuration
Authentication Proxy Banner
 HTTP Protocol Banner: Auth-Proxy-Banner-Text
Authentication global cache time is 205 minutes
Authentication global absolute time is 305 minutes
Authentication global init state time is 15 minutes
Authentication Proxy Session ratelimit is 100
Authentication Proxy Session Watch-list is enabled
Watch-list expiry timeout is 50 minutes
Authentication Proxy Auditing is enabled
Max Login attempts per user is 10
Authentication Proxy Rule Configuration
Auth-proxy name auth_rule
```

```
 http list 103 inactivity-timer 205 minutes
Router#
```

The following is sample output of the **show ip admission**command when the **cache** keyword is used. After a successful Telnet/HTTP-proxy session, from a Cisco Trust Agent (CTA) client to an Audit Server, is established, logs are displayed.

```
Router# show ip admission cache
Authentication Proxy Cache
Client Name aaatestuser, Client IP 192.168.101.131, Port 1870, timeout 205, Time Remaining
 205, state ESTAB
```

show logging

The following is sample output of the **show logging**command.

```
Router# show logging
Log Buffer (65000 bytes):
*Jul 3 05:33:13:935: %SYS-5-CONFIG_I: Configured from console by console
*Jul 3 05:33:18.471: USRGRP-API: [Type=IPv4 Val=192.168.101.131 Group-h_ug]: Usergroup
opcode entry deletion.
*Jul 3 05:33:18.471: %UG-6-MEMBERSHIP: IP=192.168.101.131| INTERFACE=Vlan|
USERGROUP=eng_group_ug| STATUS-REMOVED
*Jul 3 05:33:18.471: USRGRP-ENTRY: [Type=IPv4 Val=192.168.101.131 :: Group=eng_group_ug
Count=0]:Usergroup entry deleted
*Jul 3 05:33:18.471: USRGRP-ENTRY: [Type=IPv4 Val=192.168.101.131 :: Group=eng_group_ug
Count=0]:Usergroup entry clean up and free
*Jul 3 05:33:18.471: USRGRP-DB: Group=h_ug Count=0: Usergroup is empty. Destroy Group.
*Jul 3 05:33:18.471: USRGRP-DB: Group=h_ug Count=0: Clean up and free usergroup db.
*Jul 3 05:33:22.383: USRGRP-API: [Type=IPv4 Val=192.168.101.131 Group=eng_group_ug]: Usergroup
 opcode entry addition.
*Jul 3 05:33:22.383: USRGRP-DB: Group=h_ug Count=0 New usergroup db created.
*Jul 3 05:33:22.383: %UG-6-MEMBERSHIP: IP=192.168.101.131| INTERFACE=Vlan333|
USERGROUP=eng_group_ug| STATUS=ESTABLISHED
*Jul 3 05:33:22.383: USRGRP-ENTRY: [Type=IPv4 Val=192.168.101.131 :: Group=eng_group_ug
Count=1]: Usergroup entry added
*Jul 3 05:33:41.239: USRGRP-API: [Type=IPv4 Val=192.168.101.131 Group=eng_group_ug]: Usergroup
 opcode entry deletion.
*Jul 3 05:33:41.239: %UG-6-MEMBERSHIP: IP=192.168.101.131| INTERFACE=Vlan333|
USERGROUP=eng_group_ug| STATUS=REMOVED
*Jul 3 05:33:41.239: USRGRP-ENTRY: [Type=IPv4 Val=192.168.101.131 :: Group=eng_group_ug
Count=0]: Usergroup entry deleted
*Jul 3 05:33:41.239: USRGRP-ENTRY: [Type=IPv4 Val=192.168.101.131 :: Group=eng_group_ug
Count=0]: Usergroup entry clean up and free
*Jul 3 05:33:41.239: USRGRP-DB: Group=eng_group_ug Count=0: Usergroup is empty. Destroy
group.
*Jul 3 05:33:41.239: USRGRP-DB: Group=eng_group_ug Count=0: Clean up and free usergroup db.
*Jul 3 05:33:50.687: USRGRP-API: {Type=IPv4 Val=192.168.101.131 Group=eng_group_ug]: Usergroup
 opcode entry addition.
*Jul 3 05:33:50.687: USRGRP-DB: Group=eng_group_ug Count=0: New usergroup db created.
*Jul 3 05:33:50.687: %UG-6-MEMBERSHIP: IP=192.168.101.131| INTERFACE=Vlan333|
USERGROUP=eng_group_ug| STATUS=ESTABLISHED
*Jul 3 05:33:50.687: USRGRP-ENTRY: [Type=IPv4 Val=192.168.101.131 :: Group=eng_group_ug
Count=1]: Usergroup entry added
```

show policy-map type inspect zone-pair

The following is sample output of the **show policy-map type inspect zone-pair**command when the **sessions** keyword is used.

```
Router# show policy-map type inspect zone-pair sessions
policy exists on zp out_in
 Zone-pair: out_in
  Service-policy inspect: all_ins_cm_pm
   Class-map: vendor_group_ins_cm (match-all)
    Match: user-group vendor_group_ug
   Class-map: manager_group_ins_cm (match-all)
    Match: protocol telnet
```

```
     Match: user-group manager_group_ug
    Class-map: auth_proxy_ins_cm (match-all)
     Match: user-group auth_proxy_ug
     Match: protocol telnet
     Number of Established Sessions = 1
     Established Sessions
      Session 49D12BE0 (192.168.101.131:1872)=>(192.168.104.136:23) telnet:tcp SIS_OPEN
       Created 00:00:15, Last heard 00:00:09
       Bytes sent (initiator:responder) [171:249]
    Class-map: eng_group_ins_cm (match-all)
     Match: user-group eng_group_ug
     Match: protocol ftp
     Number of Established Sessions = 1
     Established Sessions
      Session 49D12E20 (192.168.101.131:1874)=>(192.168.104.136:21) ftp:tcp SIS_OPEN
       Created 00:00:12, Last heard 00:00:06
       Bytes sent (initiator:responder) [45:137]
    Class-map: sales_group_ins_cm (match-all)
     Match: protocol ftp
     Match: user-group sales_group_ug
    Class-map: class-default (match-any)
     Match: any
```

show user-group

The following is sample output of the **show user-group**command when the **configuration** keyword is used.

```
Router# show user-group
Usergroup: auth_proxy_ug
----------------------------------------------------------
User Name    Type  Interface   Learn  Age (min)
----------------------------------------------------------
192.168.101.131 IPv4     Vlan333   Dynamic  0
Usergroup: eng_group_ug
----------------------------------------------------------
User Name    Type  Interface   Learn  Age (min)
----------------------------------------------------------
192.168.101.131 IPv4     Vlan333   Dynamic  0
```

The following is sample output of the **show user-group**command when the *group-name* argument is used.

```
Router# show user-group auth_proxy_ug
Usergroup: auth_proxy_ug
----------------------------------------------------------
User Name    Type  Interface   Learn  Age (min)
----------------------------------------------------------
192.168.101.131 IPv4     Vlan333   Dynamic  0
```

The following is sample output of the **show user-group**command when the **count**keyword is used.

```
Router# show user-group count
Total Usergroup: 2
----------------------------------------------------------
User Group    Members
----------------------------------------------------------
auth_proxy_ug    1
eng_proxy_ug    1
```

Configuration Examples for User-Based Firewall Support

Cisco IOS Authentication Proxy Example

The following example shows how to configure User-Based Firewall Support. The Cisco IOS Authentication Proxy maps two users to different user-groups. Zone Policy Firewall policies are configured on a per user-group basis.

```
!IP Admission configuration
Configure the rule for HTTP based proxy authentication and associate the control plane tag
 service policy.
!
configure terminal
ip admission name auth-http proxy http service-policy type tag global-policy
ip http server
ip http secure-server
!AAA configuration
!
aaa new-model
!
aaa authentication login default group radius
aaa authentication login noAAA none
aaa authentication eou default group radius
aaa authorization network default group radius local
aaa authorization auth-proxy default group radius
aaa accounting auth-proxy default start-stop group radius
aaa accounting system default start-stop group radius
aaa session-id common
!
radius-server attribute 6 on-for-login-auth
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
radius-server configure-nas
radius-server host 192.168.104.131 auth-port 1645 acct-port 1646 key cisco
radius-server host 192.168.104.132 auth-port 1645 acct-port 1646 key cisco
radius-server source-ports extended
radius-server vsa send authentication
!
!Tag and Template configuration.
Configuration policy attributes for the engineer.
!
identity policy engineer-policy
 access-group engineer-acl
 user-group group-engineer
identity policy manager-policy
 access-group manager-acl
 user-group group-manager
!Define type control tag class-maps
!
class-map type control tag match-all auth_proxy_tag_cm
match tag auth_proxy_tag
class-map type control tag match-all eng_tag_cm
match tag eng_group_tag
class-map type control tag match-all manager_tag_cm
match tag manager_group_tag
!
!Define the control plane tag policy map.
!
policy-map type tag control tag global-policy
 class engineer-class
  identity policy engineer-policy
 class manager-class
  identity policy manager-policy
!Define per-user group traffic classification based on membership of the source IP address
 in the specified user-group.
```

```
!
class-map type inspect match-all engineer-insp-cmap
 match user-group group-engineer
 match protocol tcp
 match protocol udp
class-map type inspect match-all manager-insp-cmap
 match user-group group-manager
 match protocol http
!Zone Policy Firewall configuration.
Configure zones z1 and z2.
!
zone security z1
zone security z2
!Configure the policy map to inspect traffic between z1 and z2.
!
policy-map type inspect z1-z2-policy
 class type inspect engineer-insp-cmap
  inspect
 class type inspect manager-insp-cmap
  inspect
!Configure interfaces to their respective zones and apply the ip admission rule on the
source zone member(s).
!
interface e0
 ip admission auth-http
 zone-member security z1
interface e1
 zone-member security z2
!Configure the zone-pair and apply the appropriate policy-map.
!
zone-pair security z1-z2 source z1 destination z2
 service-policy type inspect z1-z2-policy
```

Additional References

The following sections provide references related to the User-Based Firewall Support feature.

Related Documents

Related Topic	Document Title
Cisco IOS Firewall Design	The Cisco IOS Firewall Design Guide
Cisco IOS firewall commands	*Cisco IOS Security Command Reference*
Cisco IOS Tag and Template	“Tag and Template” module
Cisco IOS Zone-Based Policy Firewall	Zone-Based Policy Firewall” module
Cisco IOS Authentication Proxy	“Authentication Proxy” module

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

Feature Information for User-Based Firewall Support

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

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

Table 27: Feature Information for User-Based Firewall Support

Feature Name	Releases	Feature Information
User-Based Firewall Support	12.4(20)T	This feature provides the option for configuring a security solution to dynamically authenticate and enforce policies on a per user basis in Cisco IOS software for Release 12.4(20)T and later releases. In Release 12.4(20)T, this feature was introduced on the Cisco 7200, Cisco 1800, Cisco 2800, and Cisco 3800 routers. The following commands were introduced or modified: **debug user-group**, **match user-group**, **show debugging**, **show user-group**, **user-group**, **user-group logging**.
LDAP Active Directory support for authproxy	15.1(1)T	This feature enables the authentication proxy to authenticate and authorize the users with the Active Directory server using LDAP. The following commands were introduced or modified: **aaa authentication** , **aaa authorization**, **attribute map**, **bind authenticate**, **base-dn**, **ipv4**, **ldap attribute map**, **map type**, **ldap server**.

CHAPTER 15

On-Device Management for Security Features

The On-Device Management for Security Features provides an intuitive and simple management interface, the Cisco Configuration Professional Express, to deploy a variety of security features. The security features available through the Cisco Configuration Professional Express are zone-based firewalls, VPN, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), URL filtering, and content scan.

The Cisco Configuration Professional Express uses existing zone-based firewall CLIs in conjunction with Network-Based Application Recognition 2 (NBAR2) CLIs to determine the application category, and position NBAR2 protocols supported by the firewall into the relevant application category.

This module provides a brief overview of the feature and describes in detail the enablement of NBAR2 for zone-based firewalls.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Information About On-Device Management for Security Features

On-Device Management for Security Features Overview

The following features are available in the Cisco Configuration Professional Express for the on-device management of security features:

- Displays the default zone-based firewall policy assignment; the policy between the LAN zone and WAN zone.
- Configures other firewall policies, in addition to default firewall policy.
- Displays default zones (the LAN zone and WAN zone)
- Assigns or removes interfaces to/from a zone.
- Creates and customizes zones.
- Displays the default Intrusion Prevention System (IPS) configuration.
- Provides a knob to enable or disable IPS globally.
- Validates the IPS master signature file; cisco public key.
- Lists IPS signatures in use.
- Configures and manages filtering for specific domains or websites.
- Provides a listing of popular domains that are intended to be blocked.
- Informs users when their access to domains and websites are blocked.
- Provides filtering of HTTP and Secure HTTP (HTTPS)-based access to domains.

NBAR2 Enablement in Zone-Based Firewalls

In Cisco IOS Release 15.5(1)T and later releases, zone-based firewalls supports Network-Based Application Recognition 2 (NBAR2).

NBAR2, is a classification engine that recognizes and classifies a wide variety of protocols and applications, including web-based and other difficult-to-classify applications and protocols that use dynamic TCP/UDP port assignments. With the NBAR2, enablement in zone-based firewalls, the traffic flow classification is done by NBAR2.

NBAR2 classification of traffic flows happens once and the classification results are used by multiple features including the firewall; thus avoiding flow classification by multiple features and saving router resources. NBAR2 keeps updating the Protocol Description Language (PDL) to cater to new protocols and enhancements to existing protocols. With NBAR2 enablement, the firewall does not need to update application layer gateways (ALGs).

NBAR2 Protocol Signatures Overview

NBAR2 protocol descriptions are written in StILE (Stateful Inspection Language Engine) and NBAR2 signatures are written into Protocol Description Language (PDL) files, which have a .PDL extension. Typically, each protocol has one .PDL file. Each PDL has a set of handlers that define match conditions, such as well-known port, the regular expression available in a packet, and so on. Further checks to strengthen signatures can be added within the handlers. Port-based and regular expression-based match conditions are together termed as heuristics in NBAR2 terminology. NBAR2 supports dynamic loading of PDLs which define new protocols or update existing protocols.

The following match conditions are supported by the NBAR2 Enablement in Zone-Based Firewalls feature:

- Port-based: Based on the TCP or UDP port on which a packet is available.
- Regular expression or pattern-based: Based on a specific regular expression, or fixed patterns at specific offsets found in a packet. This check can be done on the first data packet of a traffic flow in either direction, or on all packets of a flow till the classification is successful.
- General: Based on general hooks; where all packets in a flow are checked until the classification is successful.

The following are some of the key functionalities of NBAR2:

- Matches protocol-specific fields for classification of packets.
- Uses derived flows (example, FTP data flows) that are based on application-specific information derived from packets.
- Flow table manipulation based on entries in the global flow cache. These entries are added, deleted or modified by using specific PDL constructs. These entries are directional, and typically, either half-tuple or full tuple-based.
- Dynamic CLI generation based on the **match protocol** *protocol-name* command for dynamically generating the protocol name options and other NBAR2 commands.
- Subport classification (or subclassification) based on the characteristics of a protocol, such as HTTP headers (example, URL, and host), or Citrix priority tags. A PDL that provides subclassification, can specify the set and type of parameters that it supports. Subport classification also results in the generation of dynamic CLI generation.
- Maintaining of cross-packet states using local tables.
- Classification of tunneled protocols (example, Yahoo messenger over HTTP).
- Limited support for field extraction.

How to Configure On-Device Management for Security Features

Enabling NBAR2 in Zone-Based Firewalls

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **nbar-classify**
5. **end**
6. **show parameter-map type inspect global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Configures a global inspect type parameter map and enters parameter-map inspect configuration mode.
Step 4	**nbar-classify** **Example:** `Device(config-profile)# nbar-classify`	Configures Network-Based Application Recognition 2 (NBAR2) classification for the zone-based firewall inspection.
Step 5	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map inspect configuration mode and returns to privileged EXEC mode.
Step 6	**show parameter-map type inspect global** **Example:** `Device# show parameter-map type inspect global`	Displays the global inspect type parameter map values.

The following sample output from the **show parameter-map type inspect global** command displays the NBAR2 configuration along with configurations available in the global parameter map:

```
Device# show parameter-map type inspect global

alert on
  sessions maximum 2147483647
  waas disabled
  l2-transparent dhcp-passthrough disabled
  log dropped-packets disabled
  log summary disabled
  max-incomplete low 18000
  max-incomplete high 20000
  one-minute low 2147483647
  one-minute high 2147483647
  tcp reset-PSH disabled
  exporter not-configured
  nbar-classify
```

Configuring NBAR2 Protocols in a Class Map

To enable web application traffic such as Facebook, Twitter, LinkedIn and so on, you must enable basic web application protocols such as HTTP, Secure HTTP (HTTPS), or Domain Name System (DNS) to inspect traffic. For example, to enable facebook traffic, you must enable either HTTP, HTTPS or DNS traffic. When the web application uses the well-known port of a protocol; for example, port 80 that is assigned to HTTP, the initial traffic session is classified as HTTP protocol, based on the Layer 4 port. However, if subsequent packets match the web application protocol signature, the session is reclassified as the web application protocol.

```
class-map type inspect c1
 match protocol http
 pass
!
class-map type inspect c2
 match protocol facebook
 drop
!
 class-default
  drop
```

Multiple classes are needed to drop traffic from a web application, and inspect or pass the remaining traffic. The web application desired to be dropped needs to be set to drop in a separate class. In the configuration example below, if NBAR classifies traffic as "twitter"/"linkedin" firewall hits the class-default. In class-default if parent protocol is set to pass it will continue to do the parent class action, instead of dropping the packet. To explicitly drop, user should add drop action for each protocol need to be dropped.

You must remove the NBAR2 protocol match statements from the class map, before you disable NBAR2 using the **no nbar-classify** command.

Before You Begin

Prerequisites
You must enable Network-Based Application Recognition 2 (NBAR2) for zone-based firewalls by using the **nbar-classify** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **class-map type inspect** *class-map-name*
7. **match protocol** *facebook*
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** *class-map-name*
11. **pass**
12. **exit**
13. **class type inspect** *class-map-name*
14. **drop**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect** *class-map-name* **Example:** `Device(config)# class-map type inspect cmap1`	Specifies the traffic class on which an action is to be performed and enters the class map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol http`	Configures a match criterion for a class map on the basis of a specified protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class map configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	**class-map type inspect** *class-map-name* **Example:** `Device(config)# class-map type inspect cmap-new`	Specifies the traffic class on which an action is to be performed and enters the class map configuration mode.
Step 7	**match protocol** *facebook* **Example:** `Device(config-cmap)# match protocol facebook`	Configures a match criterion for a class map on the basis of a specified protocol.
Step 8	**exit** **Example:** `Device(config-cmap)# exit`	Exits class map configuration mode and returns to global configuration mode.
Step 9	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect pmap1`	Creates a Layer 3 and Layer 4 inspect type policy map and enters policy map configuration mode.
Step 10	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect cmap1`	Specifies the traffic class on which an action is to be performed and enters the policy-map class configuration mode.
Step 11	**pass** **Example:** `Device(config-pmap-c)# pass`	Allows packets to be sent to the device without firewall inspection.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy map configuration mode.
Step 13	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect cmap-new`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 14	**drop** **Example:** `Device(config-pmap-c)# drop`	Configures a traffic class to discard packets that belong to a specific class.
Step 15	**end** **Example:** `Device(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Configuration Examples for On-Device Management for Security Features

Example: Enabling NBAR2 in Zone-Based Firewalls

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# nbar-classify
Device(config-profile)# end
```

Example: Configuring NBAR2 Protocols in a Class Map

```
Device# configure terminal
Device(config)# class-map type inspect cmap1
Device(config-cmap)# match protocol http
Device(config-cmap)# exit
Device(config)# class-map type inspect cmap-new
Device(config-cmap)# match protocol facebook
Device(config-cmap)# exit
Device(config)# policy-map type inspect pmap1
Device(config-pmap)# class type inspect cmap1
Device(config-pmap-c)# pass
Device(config-pmap-c)# exit
Device(config-pmap)# class type inspect cmap-new
Device(config-pmap-c)# drop
Device(config-pmap-c)# end
```

Additional References for On-Device Management for Security Features

Related Documents

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

Related Topic	Document Title
Cisco Configuration Professional	http://www.cisco.com/c/en/us/support/cloud-systems-management/configuration-professional/tsd-products-support-series-home.html

Technical Assistance

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

Feature Information for On-Device Management for Security Features

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

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

Table 28: Feature Information for On-Device Management for Security Features

Feature Name	Releases	Feature Information
On-Device Management for Security Features	Cisco IOS Release 15.5(1)T	The On-Device Management for Security Features provides an intuitive and simple management interface, the Cisco Configuration Professional Express, to deploy a variety of security features. The security features available through the Cisco Configuration Professional Express are zone-based firewalls, VPN, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), and URL filtering. This module provides a brief overview of the feature and describes how to enable NBAR2 for zone-based firewalls. The following commands were introduced or updated for this feature: **nbar-classify** and **show parameter-map type inspect global**.