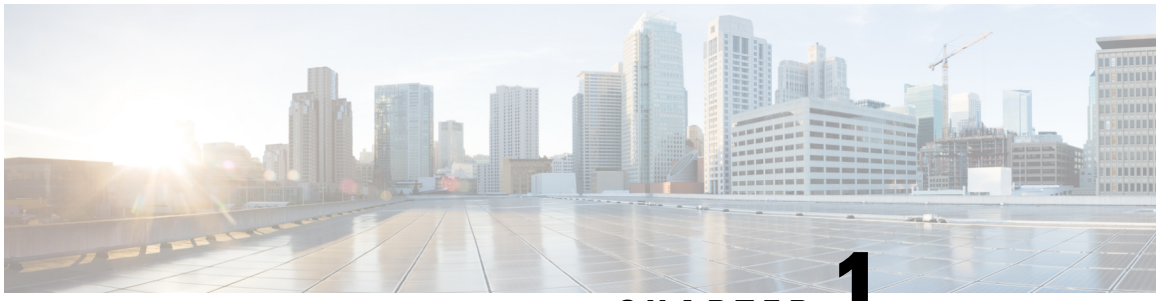
Security Configuration Guide: Zone-Based Policy Firewall, Cisco IOS XE Fuji 16.9.x

Last Modified: 2018-08-31

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

CONTENTS



CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E for Catalyst Switching and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

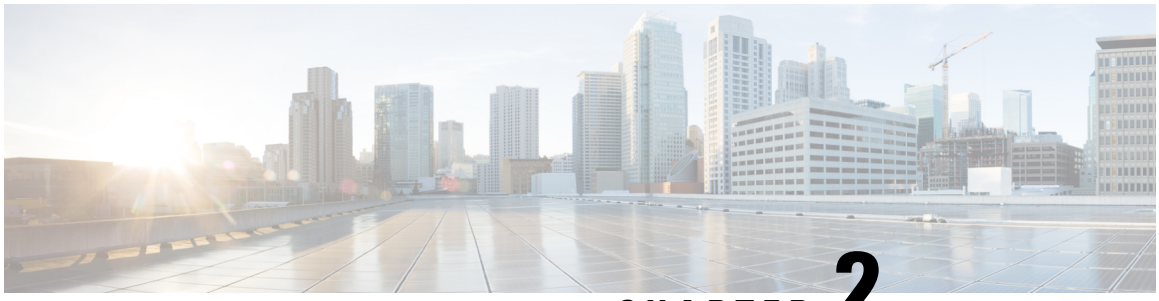
Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.



CHAPTER 2

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 Firewalls

Before you create zones, you should group interfaces that are similar when they are viewed from a security perspective.

Restrictions for Zone-Based Policy Firewalls

- In a Cisco Wide Area Application Services (WAAS) and Cisco IOS XE firewall configuration, all packets processed by a Wide Area Application Engine (WAE) device must go over the Cisco IOS XE firewall in both directions to support the Web Cache Coordination Protocol (WCCP) generic routing encapsulation (GRE) redirect. This situation occurs when a Layer 2 redirect is not available. If a Layer 2 redirect is configured on the WAE, the system defaults to the GRE redirect to continue to function.
- The zone-based firewall cannot interoperate with WAAS and WCCP, when WCCP is configured with Layer 2 redirect method.
- Zone-based Firewall configuration cannot be applied on Bridge Domain Interfaces (BDI) that involves a vCUE call flow.
- The self zone is the only exception to the default deny all policy. All traffic to any router interface is allowed until traffic is explicitly denied.
- In a WAAS and Cisco IOS XE firewall configuration, WCCP does not support traffic redirection using policy-based routing (PBR).
- WCCP traffic redirection does not work when zone-based policy firewall enabled with generic GRE is configured on a Cisco Aggregation Services Router that is configured with Cisco ISR-WAAS I/O modules. It is a Wide-Area Networking optimization solution. For WCCP traffic redirection to work, remove the zone-based policy firewall configuration from interfaces. If you are using a WAE device, WCCP traffic redirection works correctly.

  In the context of WAAS, generic GRE is an out-of-path deployment mechanism that helps to return packets from the WAAS WAE, through the GRE tunnel to the same device from which they were originally redirected, after completing optimization.
- Stateful inspection support for multicast traffic is not supported between any zones, including the self zone. Use Control Plane Policing for protection of the control plane against multicast traffic.
- 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 using the **icmp time-exceeded** and **icmp host unreachable** commands with the **pass** command (not the **inspect** command). This restriction applies to Cisco IOS XE Release 3.1S and previous releases.
- Access control lists (ACLs) in a class map is supported. However, the ACL based packet count is disabled by default. Perfilter statistics is available in zone-based firewalls from Cisco IOS XE Release 3.13S and later releases.
- Access control lists (ACLs) statements using object groups are ignored for packets that are sent to rendezvous point (RP) for processing.
- Bridge domain interfaces do not support zone-based firewall inspection, including all Layer 4 and Layer 7 inspection.
- The ZBF cannot inspect traffic when NAT NVI is enabled on the device.
- When traffic enters a zone pair, the firewall examines the entire connection table and matches the traffic with any connection in the table even if the ingress interface does not match the zone pair. In this scenario, asymmetrically routed traffic on the firewall may drop packets, if the **inspect**action is configured.

In Cisco IOS XE Release 3.15S and later releases, zone-mismatch drop is configured in the class parameter map. If **zone-mismatch drop** is set, then the zones are checked against the original zones used when the packet is classified. If the zone is not part of the zone pair, the packet is dropped. If **zone-mismatch drop** is not set, then the zones are not checked.

- When ZBF is configured, all interfaces that are a part of a zone pair much have rii configured. Interfaces that match the peer device must have the same rii configured. Additionally, flows that are initiated between two interfaces and either of them does not have an RII assigned, it does not sync to the standby.
- The zone-based firewall is supported with dynamic interfaces only in the default zone. These interfaces are created or deleted dynamically when traffic is tunneled into tunnels such as IPsec or VPN secure tunnels. Virtual templates are used to support certain types of dynamic interfaces. For more information, see Virtual Interfaces as Members of Security Zones, on page 8.
- To disable the zone-based firewall configurations that have been applied on the interfaces, use the **platform inspect disable-all**command. Similarly, to enable zone-based firewall on the interfaces, use the **no platform inspect disable-all** command.

  To verify if the **platform inspect disable-all** command has been applied, use the following show running configuration:

```
show run | sec disable
platform inspect disable-all
```

  Note By default, zone-based firewall is always enabled.

- When the **drop log** command is configured under a user-defined class or the default class of a policy, disabling the logging of dropped packets by configuring the **drop** command does not stop the log messages. This is a known issue and the workaround is to configure the **nodroplog** command before configuring the **drop** command to stop the logging of messages. This issue applies to the **pass** command as well. The following example shows the issue:

```
! Logging of dropped packets is enabled by configuring the drop log command.
policy-map type inspect INT-EXT
 class type inspect INT-EXT
  pass
 class class-default
  drop log
!
```

  The following example shows the workaround:

```
! In this example, the no drop log command is configured before the drop command.
policy-map type inspect INT-EXT
 class type inspect INT-EXT
  pass
 class class-default
  drop log
  no drop log
  drop
!
```

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. This 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, and pass commands. You can attach policy maps to a target (zone pair).

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

Overview of Zones

A zone is a group of interfaces that have similar functions or features. They help you specify where a Cisco IOS XE firewall should be 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.

Note Because the Cisco IOS XE zone-based firewall is implemented as an egress feature on a zone you must match the traffic before it leaves the zone. For example, if a Dynamic Multipoint VPN (DMVPN) tunnel terminates on the outside zone, you must allow generic routing encapsulation (GRE) traffic into the router through the zone pair that connects the outside zone with the self zone, because packets are decrypted before the firewall checks the traffic.

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.

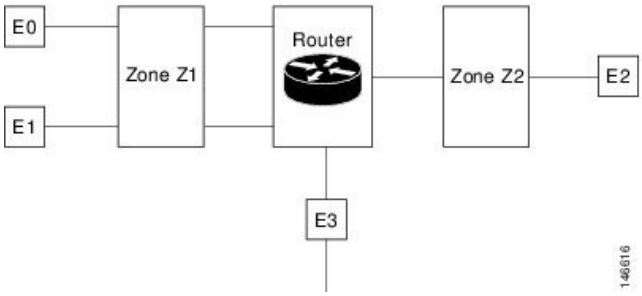
The following are basic rules to consider when setting up zones:

- 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, pass or drop the traffic between the two zones.
- 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.
- For traffic to flow among all interfaces in a device, these interfaces must be members of one security zone or another. It is not necessary for all device interfaces to be members of security zones.
- All interfaces associated with a zone must be contained in the same VRF (Virtual Routing Forwarding).

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. Traffic flows freely between interfaces E0 and E1 because they are members of the same security zone (Z1).
- If no policies are configured, traffic will not flow between any other interfaces (for example, E0 and E2, E1 and E2, E3 and E1, and E3 and E2).

- Traffic can flow between E0 or E1 and E2 only when an explicit policy permitting traffic is configured between zone Z1 and zone Z2.
- Traffic can never flow between E3 and E0/E1/E2 unless default zones are enabled.

Note On the Cisco ASR 1000 Series Aggregation Services Routers the firewall supports a maximum of 4000 zones.

Overview of Security Zone Firewall Policies

A class identifies 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 functionality that is typically associated with a traffic class. Firewall supports the following type of actions:

inspect — once classified, firewall session is created in the connection table and the packets content is examined.

pass — the packet is simply classified and the traffic is allowed to pass through the system without further inspection.

drop — the packet is classified and dropped.

To create security zone firewall policies, you must 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 that arrive at targets (such as the input interface, output interface, or zone pair), 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.

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.

For more information on the Per Subscriber Firewall on LNS feature, see the Release Notes for Cisco ASR 1000 Series Aggregation Services Routers for Cisco IOS XE Release 2.

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.

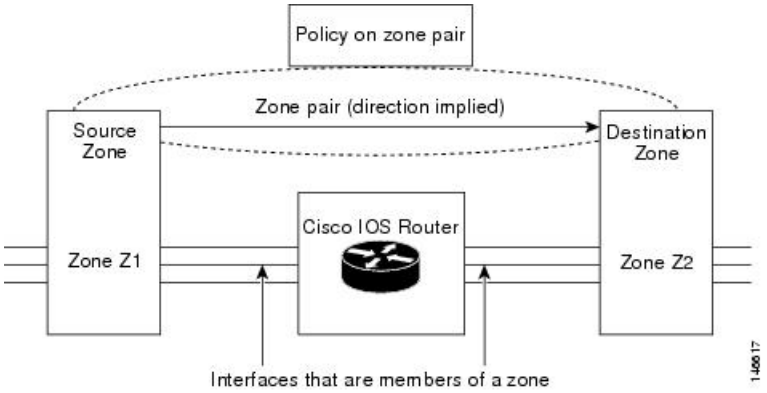
The default zone is applicable to interfaces where no security zone is associated. Default zones are by default not enabled. To enable default zones use the **zone security default** configuration command to create the default zone.

The most common usage of firewall is to apply them to traffic through a device, so you need at least two zones. For traffic to and from the device, ZBF supports the concept of a self-zone.

To permit traffic between zone member interfaces, you must configure a policy permitting (inspecting or passing) 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 may require policies for traffic going in both directions (from Z1 to Z2 and Z2 to Z1). If traffic is initiated from either direction, you must configure two zone pairs.

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 initiator 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. For the pass action, a policy must exist for packets in each direction and for inspect a policy need to exist for traffic from the initiator.

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.

A policy is not required for Internet Control Message Protocol (ICMP) error packets.

Note A policy is required for ICMP informational messages such as **ICMP_ECHO** (ping) for packet arriving from an initiator.

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 informational ICMP packets, such as ICMP_EHCO_REQUEST 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. Once a firewall connection is created, the packets are no longer classified. That is, if the policy map changes, the underlying connections are not noticed. As connection is not established, a mirrored policy with a pass action must be created packets in the reverse direction.

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. If a class map only contains an access list and does not contain a match protocol, then firewall attempts to match the flow protocol to known ALGs and process it as required.

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

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**, **pass** 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-cmap)# exit
Device(config)# policy-map type inspect p1
Device(config-pmap)# class type inspect c1
Device(config-pmap-c)# drop
```

Note On the Cisco ASR 1000 Series Aggregation Services Routers the firewall supports a maximum of 1000 policy maps and 8 classes inside a policy map. You can configure a maximum of 16 match statements in a class map and 1000 globally.

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.

Note Configure zone-based firewall on the device such that the TCP traffic flow does not exceed 65k in the window size.

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
```

Supported Protocols for Layer 3 and Layer 4

The following protocols are supported:

- FTP
- H.323
- Real-time Streaming Protocol (RTSP)
- SCCP (Skinny Client Control Protocol)
- Session Initiation Protocol (SIP)
- Trivial File Transfer Protocol (TFTP)
- RCMD
- Lightweight Directory Access Protocol (LDAP)
- Hypertext Transfer Protocol (HTTP)
- Domain Name System (DNS)
- Simple Mail Transfer Protocol (SMTP/ESMTP)
- Post Office Protocol 3 (POP3)
- Internet Mail Access Protocol (IMAP)
- SUN Remote Procedure Call (SUNRPC)
- GPRS Tunnel Protocol version 0/1 (GTPv1)
- GPRS Tunnel Protocol version 2 (GTPv2)
- Point to Point Tunneling Protocol (PPTP)

Access Control Lists and Class Maps

Access lists are packet-classifying mechanisms. Access lists define the actual network traffic that is permitted or denied when an ACL is applied to a specific class map. Thus, the ACL is a sequential collection of permit and deny conditions that applies to a packet. A router tests packets against the conditions set in the ACL one at a time. A deny condition is interpreted as "do not match." Packets that match a deny access control entry (ACE) cause an ACL process to terminate and the next match statement within the class to be examined.

Note You can configure the range of variables in an ACL as match criteria for a class-map. Because the firewall supports only the 5-tuple match criteria, only source address, source port, destination address, destination port and protocol match criteria are supported. Any other match criteria that is configured and accepted by the CLI, will not be supported by the firewall

Class maps are used to match a range of variables in an ACL based on the following criteria:

- If a class map does not match a permit or a deny condition, then the ACL fails.
- The match-all or match-any are applied to the match statements contained within the class map. ACLs are processed as normal and the result is used when comparing against match-all or match-any.
- If a match-all attribute is specified and any match condition, ACL, or protocol fails to match the packet, further evaluation of the current class is stopped, and the next class in the policy is examined.
- If any match in a match-any attribute succeeds, the class map criteria are met and the action defined in the policy is performed.
- If an ACL matches the match-any attribute, the firewall attempts to ascertain the Layer 7 protocol based on the destination port.

If you specify the match-all attribute in a class map, the Layer 4 match criteria (ICMP, TCP, and UDP) are set and the Layer 7 match criteria are not set. Hence, the Layer 4 inspection is performed and Layer 7 inspection is omitted.

Access lists come in different forms: standard and extended access lists. Standard access lists are defined to permit or deny an IP address or a range of IP addresses. Extended access lists define both the source and the destination IP address or an IP address range. Extended access lists can also be defined to permit or deny packets based on ICMP, TCP, and UDP protocol types and the destination port number of the packet.

The following example shows how a packet received from the IP address 10.2.3.4 is matched with the class test1. In this example, the access list 102 matches the deny condition and stops processing other entries in the access list. Because the class map is specified with a match-all attribute, the "class-map test1" match fails. However, the class map is inspected if it matches one of the protocols listed in test1 class map.

If the class map test1 had a match-any attribute (instead of match-all), then the ACL would have matched deny and failed, but then the ACL would have matched the HTTP protocol and performed the inspection using "pmap1."

```
access-list 102 deny ip 10.2.3.4 0.0.0.0 any
access-list 102 permit any any
class-map type inspect match-all test1
 match access-list 102
 match protocol http
!
class-map type inspect match-any test2
 match protocol sip
 match protocol ftp
```

```
 match protocol http
!
parameter-map type inspect pmap1
 tcp idle-time 15
!
parameter-map type inspect pmap2
 udp idle-time 3600
!
policy-map type inspect test
 class type inspect test1
  inspect pmap1
!
 class type inspect test2
  inspect pmap2
!
 class type inspect class-default
  drop log
```

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.

Define two access lists, Marketing and Engineering. Create a class-map that does a match-any on the two access groups. Then, create another class-map that includes the previous class-map with a match-all and match protocol http.

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 maps. 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. In general, mapping outside global addresses is not recommended.

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.

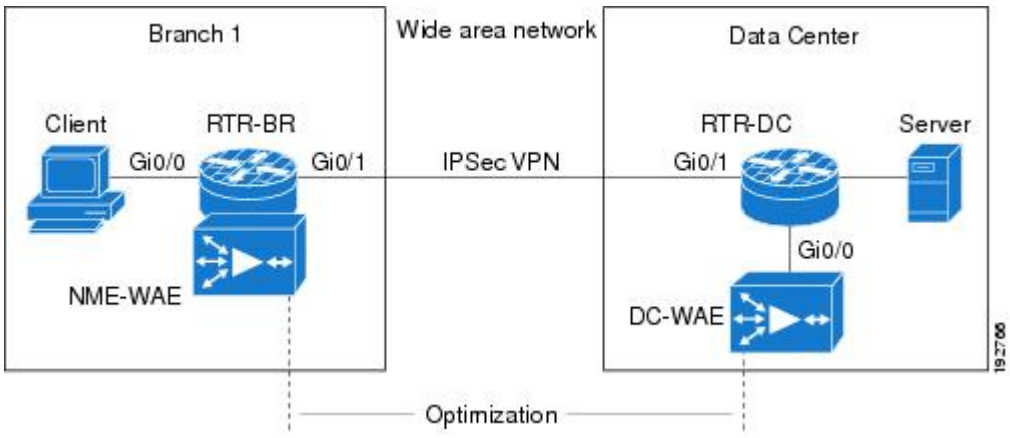
Note Stateful Layer 7 inspection on the client side can also be performed on nonoptimized traffic.

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). ZBF inspects the clear text after WAAS has unoptimized the packet.

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 3: End-to-End WAAS Optimization Path

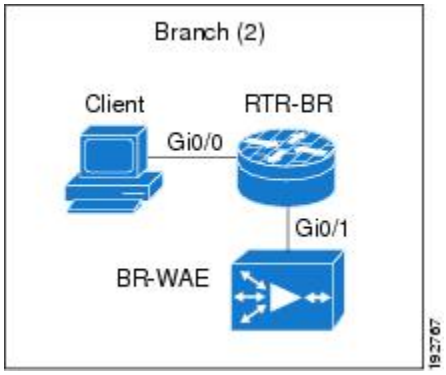


WAAS Branch Deployment with an Off-Path Device

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.

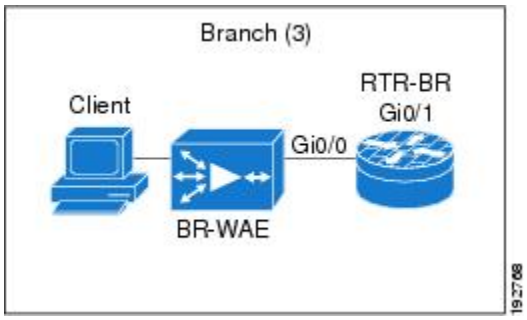
Figure 4: WAAS Off-Path Branch Deployment



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 5: 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.

Out-of-Order Packet Processing Support in the Zone-Based Firewalls

By default, the Cisco IOS XE firewall drops all out-of-order (OoO) packets when Layer 7 deep packet inspection (DPI) is enabled or when Layer 4 inspection with Layer 7 protocol match is enabled. Dropping out-of-order packets can cause significant delays in end applications because packets are dropped only after the retransmission timer expires (on behalf of the sender). Layer 7 inspection is a stateful packet inspection and it does not work when TCP packets are out of order.

In Cisco IOS XE Release 3.5S, if a session does not require DPI, OoO packets are allowed to pass through the router and reach their destination. All Layer 4 traffic with OoO packets are allowed to pass through to their destination. However, if a session requires Layer 7 inspection, OoO packets are still dropped. By not dropping OoO packets when DPI is not required, the need to retransmit dropped packets and the bandwidth needed to retransmit on the network is reduced.

Severity Levels of Debug Messages

The severity level of debug messages specifies the types of issues for which a message is logged. While enabling firewall debugging, you can specify the level of messages that should be logged. The following table provides details about severity levels of debug messages.

Table 1: Severity Levels of Firewall Debug Messages

Trace Level	Severity Levels	Description
Critical	1	Applies to issues that make the zone-based policy firewall unusable or not forward packets. This is the default. Examples of critical events are: • Back pressure triggered by the log mechanism. • Resource limit exceeded. • Memory allocation failure. • High availability state not allowing new sessions.
Error	2	Applies to all error conditions and packet-drop conditions. Examples of error events are: • Synchronized (SYN) cookie—the number of maximum destination reached. • Not an initiator packet. • Could not send packets. • Application layer gateway (ALG) error condition.

Trace Level	Severity Levels	Description
Information	3	Applies to informational messages. Examples of information events are: • Packet drop due to incorrect policy configuration, zone-check failure, malformed packets, or hardcoded limit or threshold. • State machine transition. • Session or imprecise channel database information, search results and so on. • Packet classification status or result. • Packet pass or drop status. • Session hit or miss. • Packet that is sent is a TCP reset (RST) packet. • SYN cookie event.
Detail	4	All log messages are printed. Examples of detailed events are: • Data structures. • Ternary content-addressable memory (TCAM) search keys and result structure. • Firewall event details.

Smart Licensing Support for Zone-Based Policy Firewall

Zone-Based Policy Firewall features for Cisco ASR 1000 Series Aggregation Services Routers are packaged separately from the security package and hence Zone-Based Policy Firewall requires separate license to enable and disable features. The Smart License support for Zone Based Firewall on ASR1000 feature implements support for smart licensing at a feature level for on Cisco ASR 1000 Series Aggregation Services Routers via the Universal K9 software image.

The device need not be reloaded to enable this feature. Smart licensing is not turned on by default. Smart Licensing is toggled on or off globally via the **license smart enable** command or when configuring a Zone-Based Policy Firewall via the **zone security** command. The **show license all** command displays the status of smart license when smart licensing is implemented. The following is a sample output from the **show license all** command when smart licensing is enabled globally.

```
Device# show license all

License Store: Primary License Storage
StoreIndex: 0   Feature: internal_service                Version: 1.0
        License Type: Evaluation
        License State: Active, In Use
            Evaluation total period: 1  day  0  hour
            Evaluation period left: 18 hours 57 minutes
```

```
            Period used: 5  hours 2  minutes
            Expiry date: Mar 18 2016 14:15:02
        License Count: Non-Counted
        License Priority: Low
License Store: Built-In License Storage
StoreIndex: 0   Feature: adventerprise                    Version: 1.0
        License Type: EvalRightToUse
        License State: Active, In Use
            Evaluation total period: 8  weeks 4  days
            Evaluation period left: 8  weeks 3  days
            Period used: 5  hours 13 minutes
            Transition date: May 16 2016 14:03:52
        License Count: Non-Counted
        License Priority: Low          <-- (CSL mode license)

Device(config)# license smart enable
Device(config)# zone security z1
Device(config)# exit
Device# show license all

Smart Licensing Status
----------------------------
Smart Licensing is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 65 days, 14 hours, 19 minutes, 47 seconds

License Usage
--------------

(ASR_1000_AdvEnterprise):
  Description:
  Count: 1
  Version: 1.0
  Status: EVAL MODE

(ASR_1000_firewall):
  Description:
  Count: 1
  Version: 1.0
  Status: EVAL MODE

Product Information
----------------------
UDI: PID:ASR1013,SN:NWG165000A9

Agent Version
---------------
Smart Agent for Licensing: 1.5.1_rel/29
Component Versions: SA:(1_3_dev)1.0.15, SI:(dev22)1.2.1, CH:(rel5)1.0.3, PK:(dev18)1.0.3
```

The following is a sample output when smart licensing is disabled.

```
Device(config)# no zone security z1
Device(config)# exit
Device# show license all

Smart Licensing Status
----------------------
```

```
Smart Licensing is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 65 days, 14 hours, 18 minutes, 58 seconds

License Usage
--------------

(ASR_1000_AdvEnterprise):
  Description:
  Count: 1
  Version: 1.0
  Status: EVAL MODE

Product Information
--------------------
UDI: PID:ASR1013,SN:NWG165000A9

Agent Version
-------------
Smart Agent for Licensing: 1.5.1_rel/29
Component Versions: SA:(1_3_dev)1.0.15, SI:(dev22)1.2.1, CH:(rel5)1.0.3, PK:(dev18)1.0.3

Device(config)# no license smart enable
Device(config)# exit
Device# show license all

License Store: Primary License Storage
StoreIndex: 0   Feature: internal_service                 Version: 1.0
        License Type: Evaluation
        License State: Active, Not in Use, EULA accepted
            Evaluation total period: 1  day  0  hour
            Evaluation period left: 18 hours 54 minutes
            Period used: 5  hours 5  minutes
        License Count: Non-Counted
        License Priority: Low
License Store: Built-In License Storage
StoreIndex: 0   Feature: adventerprise                    Version: 1.0
        License Type: EvalRightToUse
        License State: Active, Not in Use, EULA accepted
            Evaluation total period: 8  weeks 4  days
            Evaluation period left: 8  weeks 3  days
            Period used: 5  hours 17 minutes
        License Count: Non-Counted
        License Priority: Low               <--- (back to CSL mode)
```

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.
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:**	Configures the match criterion for a class map on the basis of a specified protocol.

	Command or Action	Purpose
	`Device(config-cmap)# match protocol http`	• Only Cisco stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
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.
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 mutually exclusive; that is, you cannot specify both of them at the same time. Only one can be specified.
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.

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*]} **Example:** `Device(config-profile)# log summary flows 15 time-interval 30`	(Optional) Configures packet logging during the firewall activity. **Note** This command is visible in parameter map type inspect configuration mode only.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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 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.

A zone pair can have the same zone for source and destination zone. By default, traffic staying within a zone is not inspected. In addition, there is the default zone (interfaces with no zone assignment) which can also be specified.

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. **interface** *type number*
7. **zone-member security** *zone-name*
8. **exit**
9. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self** | *default*] **destination** [**self** | *default* | *destination-zone-name*]
10. **description** *line-of-description*
11. **service-policy type inspect** *policy-map-name*
12. **platform inspect match-statistics per-filter**
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	**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.
Step 5	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 7	**zone-member security** *zone-name* **Example:**	Assigns an interface to a specified security zone.

	Command or Action	Purpose
	`Device(config-if)# zone-member security zone1`	**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 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self** | *default*] **destination** [**self** | *default* | *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 10	**description** *line-of-description* **Example:** `Device(config-sec-zone-pair)# description accounting network to internet`	(Optional) Describes the zone pair.
Step 11	**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 12	**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 13	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to privileged EXEC mode.

Configuring NetFlow Event Logging

Global parameter maps are used for NetFlow event logging. With NetFlow event logging enabled, logs are sent to an off-box, high-speed log collector. By default, this functionality is not enabled. (If this functionality is not enabled, firewall logs are sent to a logger buffer located in the Route Processor or console.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-global**
4. **log dropped-packets**
5. **log flow-export v9 udp destination** *ipv4-address port*
6. **log flow-export template timeout-rate** *seconds*
7. **end**
8. **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 parameter map and enters parameter-map type inspect configuration mode.
Step 4	**log dropped-packets** **Example:** `Device(config-profile)# log dropped-packets`	Enables logging for all packets dropped by firewall.
Step 5	**log flow-export v9 udp destination** *ipv4-address port* **Example:** `Device(config-profile)# log flow-export v9 udp destination 192.0.2.0 5000`	Enables NetFlow event logging and provides the collector's IP address and port.
Step 6	**log flow-export template timeout-rate** *seconds* **Example:** `Device(config-profile)# log flow-export template timeout-rate 5000`	Specifies the template timeout value.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**show parameter-map type inspect-global** **Example:** `Device# show parameter-map type inspect-global`	Displays global inspect-type parameter map information.

Configuring the Firewall with WAAS

Perform the following task to configure an end-to-end Wide Area Application Services (WAAS) traffic flow optimization for the firewall that uses) to redirect traffic to a Wide Area Application Engine (WAE) device for traffic interception. When configuring WCCP in ZBFW enviornment, pay attention using L2 redirection as GRE is required for zone based firewall.

Note Configuring the firewall with WAAS (steps 5 to 13) is not required post Cisco IOS XE Release 3.5S. The commands in steps 5 to 12 have been deprecated post Cisco IOS XE Release 3.5S.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp** *service-id*
4. **ip wccp** *service-id*
5. **log dropped-packets enable**
6. **max-incomplete low**
7. **max-incomplete high**
8. **class-map type inspect** *class-name*
9. **match protocol** *protocol-name* [**signature**]
10. **exit**
11. **policy-map type inspect** *policy-map-name*
12. **class class-default**
13. **class-map type inspect** *class-name*
14. **inspect**
15. **exit**
16. **exit**
17. **zone security** *zone-name*
18. **description** *line-of-description*
19. **exit**
20. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
21. **description** *line-of-description*

22. **exit**
23. **interface** *type number*
24. **description** *line-of-description*
25. **zone-member security** *zone-name*
26. **ip address** *ip-address*
27. **ip wccp** *service-id* {**group-listen** | **redirect** {**in** | **out**}}
28. **exit**
29. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**} **destination** [**self** | *destination-zone-name*]
30. **service-policy type inspect** *policy-map-name*
31. **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 Web Cache Communication Protocol (WCCP) dynamically defined service identifier number.
Step 5	**log dropped-packets enable** **Example:** `Device(config-profile)# log dropped-packets enable`	
Step 6	**max-incomplete low** **Example:** `Device(config)# max-incomplete low 18000`	
Step 7	**max-incomplete high** **Example:** `Device(config)# max-incomplete high 20000`	
Step 8	**class-map type inspect** *class-name* **Example:**	Creates an inspect type class map for the traffic class and enters class-map configuration mode.

	Command or Action	Purpose
	`Device(config)# class-map type inspect most-traffic`	**Note** The **class-map type inspect most-traffic** command is hidden.
Step 9	**match protocol** *protocol-name* [**signature**] **Example:** `Device(config-cmap)# match protocol http`	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 10	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
Step 11	**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 12	**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 13	**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 14	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables Cisco stateful packet inspection.
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 zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 18	**description** *line-of-description* **Example:**	(Optional) Describes the zone.

	Command or Action	Purpose
	`Device(config-sec-zone)# description Internet Traffic`	
Step 19	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 20	**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 configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 21	**description** *line-of-description* **Example:** `Device(config-sec-zone)# description accounting network`	(Optional) Describes the zone pair.
Step 22	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 23	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters interface configuration mode.
Step 24	**description** *line-of-description* **Example:** `Device(config-if)# description zone interface`	(Optional) Describes an interface.
Step 25	**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 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 26	**ip address** *ip-address* **Example:** `Device(config-if)# ip address 10.70.0.1 255.255.255.0`	Assigns an interface IP address for the security zone.

	Command or Action	Purpose
Step 27	**ip wccp** *service-id* {**group-listen** | **redirect** {**in** | **out**}} **Example:** `Device(config-if)# ip wccp 61 redirect in`	Specifies WCCP parameters on the interface.
Step 28	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 29	**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.
Step 30	**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 31	**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: Creating an Inspect Parameter Map

```
parameter-map type inspect eng-network-profile
 alert on
 audit-trail on
 dns-timeout 60
 icmp idle-timeout 90
 max-incomplete low 800
 one-minute low 300
 sessions maximum 200
 tcp finwait-time 5
 tcp idle-time 90
 tcp max-incomplete host 500 block-time 10
 tcp synwait-time 3
 udp idle-time 75
```

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: 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!
```

Example: Configuring NetFlow Event Logging

```
parameter-map type inspect global
 log dropped-packets
 log flow-export v9 udp destination 192.0.2.0 5000
 log flow-export template timeout rate 5000
```

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
 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
!
```

Example: Configuring Firewall with FlexVPN and DVTI Under the Same Zone

The following example shows a firewall with FlexVPN and Dynamic Virtual Tunnel Interfaces (DVTI) configured under the same zone.

```
crypto ikev2 proposal PROP
 encryption 3des
 integrity sha256
 group 5
crypto ikev2 policy POL
 match fvrf any
 proposal PROP
crypto ikev2 keyring keyring1
 peer peer
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco
crypto ikev2 profile prof1
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback1
 keyring local keyring1
 no shutdown
 Virtual-Template 1
class-map type inspect match-any cmap
 match protocol icmp
 match protocol tcp
 match protocol udp
policy-map type inspect pmap
 class type inspect cmap
 inspect
 class class-default
 drop log
zone security in
zone security zone1
zone-pair security zp1 source zone1 destination in
 service-policy type inspect pmap
crypto ipsec profile ipsec1
 set ikev2-profile prof1
interface Loopback1
 ip address 51.1.1.1 255.255.255.0
interface Gi0/0/0.2
 encapsulation dot1q 2
 ip address 100.1.1.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.3
 encapsulation dot1q 3
 ip address 100.1.2.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.4
 encapsulation dot1q 4
 ip address 100.1.3.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.5
 encapsulation dot1q 5
 ip address 100.1.4.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.6
 encapsulation dot1q 6
 ip address 100.1.5.1 255.255.255.0
 zone-member security in
interface Virtual-Template1 type tunnel
 ip unnumbered loopback1
 zone-member security zone1
 tunnel source loopback1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec1
ip route 60.0.0.0 255.0.0.0 192.168.2.2
```

Example: Configuring Firewall with FlexVPN and DVTI Under a Different Zone

The following example shows a firewall with FlexVPN and Dynamic Virtual Tunnel Interfaces (DVTI) configured under a different zone.

```
crypto ikev2 proposal PROP
 encryption 3des
 integrity sha256
 group 5
crypto ikev2 policy POL
 match fvrf any
 proposal PROP
crypto ikev2 keyring keyring1
 peer peer1
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco1
crypto ikev2 keyring keyring2
 peer peer2
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco2
crypto ikev2 keyring keyring3
 peer peer3
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco3
crypto ikev2 keyring keyring4
 peer peer4
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco4
crypto ikev2 keyring keyring5
 peer peer5
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco5
crypto ikev2 profile prof1
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback1
 keyring local keyring1
 no shutdown
 Virtual-Template 1
crypto ikev2 profile prof2
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback2
 keyring local keyring2
 no shutdown
 Virtual-Template 2
crypto ikev2 profile prof3
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback3
 keyring local keyring3
crypto ikev2 profile prof4
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback4
 keyring local keyring4
 no shutdown
 Virtual-Template 4
crypto ikev2 profile prof5
```

```
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback5
 keyring local keyring5
 no shutdown
 Virtual-Template 5
class-map type inspect match-any cmap
 match protocol icmp
 match protocol tcp
 match protocol udp
policy-map type inspect pmap
 class type inspect cmap
 inspect
 class class-default
 drop log
zone security in
zone security zone1
zone security zone2
zone security zone3
zone security zone4
zone security zone5
zone-pair security zp1 source zone1 destination in
 service-policy type inspect pmap
zone-pair security zp2 source zone2 destination in
 service-policy type inspect pmap
zone-pair security zp3 source zone3 destination in
 service-policy type inspect pmap
zone-pair security zp4 source zone4 destination in
 service-policy type inspect pmap
zone-pair security zp5 source zone5 destination in
 service-policy type inspect pmap
crypto ipsec profile ipsec1
 set ikev2-profile prof1
crypto ipsec profile ipsec2
 set ikev2-profile prof2
crypto ipsec profile ipsec3
 set ikev2-profile prof3
crypto ipsec profile ipsec4
 set ikev2-profile prof4
crypto ipsec profile ipsec5
 set ikev2-profile prof5
interface Loopback1
 ip address 50.1.1.1 255.255.255.0
interface Loopback2
 ip address 50.1.2.1 255.255.255.0
interface Loopback3
 ip address 50.1.3.1 255.255.255.0
interface Loopback4
 ip address 50.1.4.1 255.255.255.0
interface Loopback5
 ip address 50.1.5.1 255.255.255.0
interface Gi0/0/0.2
 encapsulation dot1q 2
 ip address 100.1.1.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.3
 encapsulation dot1q 3
 ip address 100.1.2.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.4
 encapsulation dot1q 4
 ip address 100.1.3.1 255.255.255.0
 zone-member security in
```

```
interface Gi0/0/0.5
 encapsulation dot1q 5
 ip address 100.1.4.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.6
 encapsulation dot1q 6
 ip address 100.1.5.1 255.255.255.0
 zone-member security in
interface Virtual-Template1 type tunnel
 ip unnumbered loopback1
 zone-member security zone1
 tunnel source loopback1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec1
interface Virtual-Template2 type tunnel
 ip unnumbered loopback2
 zone-member security zone2
 tunnel source loopback2
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec2
interface Virtual-Template3 type tunnel
 ip unnumbered loopback3
 zone-member security zone3
 tunnel source loopback3
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec3
interface Virtual-Template4 type tunnel
 ip unnumbered loopback4
 zone-member security zone4
 tunnel source loopback4
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec4
interface Virtual-Template5 type tunnel
 ip unnumbered loopback5
 zone-member security zone5
 tunnel source loopback5
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec5
ip route 60.0.0.0 255.0.0.0 192.168.2.2
```

Additional References for Zone-Based Policy Firewalls

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

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 Policy Firewalls

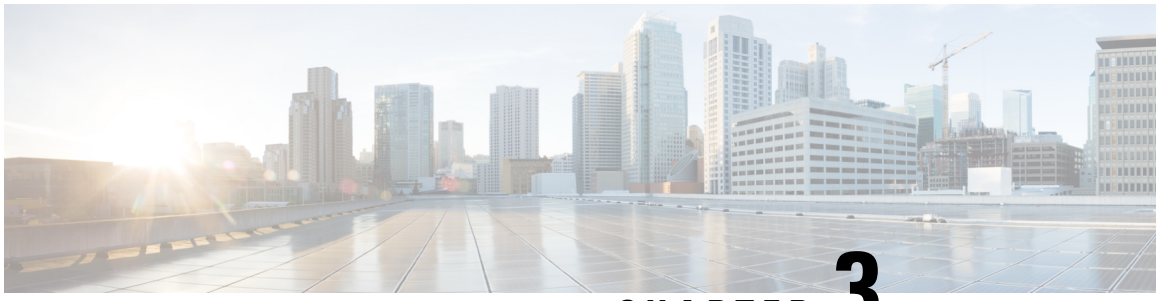
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 2: Feature Information for Zone-Based Policy Firewalls

Feature Name	Releases	Feature Information
Debuggability Enhancement in Zone Based Firewall (Phase-II)	Cisco IOS XE Release 3.10S	The Debuggability Enhancement Zone-Based Firewall provides severity levels for debug logs.
Firewall—NetMeeting Directory (LDAP) ALG Support	Cisco IOS XE Release 3.1S	LDAP is an application protocol that is used for querying and updating information stored on directory servers. The Firewall—Netmeeting Directory ALG Support feature enables Cisco firewalls to support Layer 4 LDAP inspection by default. The following command was introduced or modified by this feature: **match protocol**.
IOS-XE ZBFW interop with crypto VPN	Cisco IOS XE Release 3.17S	The IOS-XE ZBFW interop with crypto VPN feature supports enabling zone-based firewall under FlexVPN DVTI. No commands were introduced or updated by this feature.
Out-of-Order Packet Handling in Zone-Based Policy Firewall	Cisco IOS XE Release 3.5S	The Out-of-Order Packet Handling feature allows OoO packets to pass through the router and reach their destination if a session does not require DPI. All Layer 4 traffic with OoO packets are allowed to pass through to their destination. However, if a session requires Layer 7 inspection, the OoO packets are still dropped.

Feature Name	Releases	Feature Information
Smart License support for Zone Based Firewall on ASR1000	IOS XE Denali 16.3.1	Zone-Based Policy Firewall features for Cisco ASR 1000 Series Aggregation Services Routers are packaged separately from the security package and hence Zone-Based Policy Firewall requires separate license to enable and disable features. The Smart License support for Zone Based Firewall on ASR1000 feature implements support for smart licensing at a feature level for on Cisco ASR 1000 Series Aggregation Services Routers via the Universal K9 software image. The following command was modified: **show license all**.
Zone-Based Policy Firewalls	Cisco IOS Release 2.1	The Zone-Based Policy Firewall feature provides a Cisco IOS XE software unidirectional firewall policy between groups of interfaces known as zones.
Zone-Based Firewall—Default Zone	Cisco IOS Release 2.6	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 the default zone.
Zone-Based Firewall Support of Multipath TCP	Cisco IOS XE Release 3.13S	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). No commands were introduced or updated by this feature.



CHAPTER 3

Zone-Based Policy Firewall IPv6 Support

The zone-based policy firewall provides advanced traffic filtering or inspection of IPv4 packets. With IPv6 support, the zone-based policy firewall supports the inspection of IPv6 packets. Prior to IPv6 support, the firewall supported only the inspection of IPv4 packets. Only Layer 4 protocols, Internet Control Messaging Protocol (ICMP), TCP, and UDP packets are subject to IPv6 packet inspection.

This module describes the firewall features that are supported and how to configure a firewall for IPv6 packet inspection.

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 Policy Firewall IPv6 Support

The following functionalities are not supported:

- Application-level gateways (ALGs)
- Box-to-box high availability (HA)
- Distributed Denial-of-Service attacks
- Firewall resource management

- Layer 7 inspection
- Multicast packets
- Per-subscriber firewall or the broadband-based firewall
- Stateless Network Address Translation 64 (NAT64)
- VRF-Aware Software Infrastructure (VASI)
- Wide Area Application Services (WAAS) and Web Cache Communication Protocol (WCCP)

Information About IPv6 Zone-Based Firewall Support over VASI Interfaces

IPv6 Support for Firewall Features

The firewall features described in the table below are supported by IPv6 packet inspection:

Table 3: Firewall Features Supported on IPv6

Feature	Configuration Information
Class maps	*Zone-Based Policy Firewall* module.
Internet Control Message Protocol Version 6 (ICMPv6), TCP, and UDP protocols	• *Firewall Stateful Inspection of ICMP* module. • *Zone-Based Policy Firewall* module.
IP fragmentation	*Virtual Fragmentation Reassembly* module.
Intrachassis HA	—
Logging of error messages	*Zone-Based Policy Firewall* module.
Nested class maps	*Nested Class Map Support for Zone-Based Policy Firewall* module.
Out-of-order packet handling	The “Out-of-Order Packet Handling” section in the *Zone-Based Policy Firewall* module.
Parameter-maps—For inspect type parameter maps, the number of sessions defined in the parameter map will be cumulative for IPv4 and IPv6 sessions	*Zone-Based Policy Firewall* module.
Policy maps	*Zone-Based Policy Firewall* module.
Port-to-application mapping	—

Feature	Configuration Information
Stateful Network Address Translation 64 (NAT64)	The *Stateful Network Address Translation 64* module in the *IP Addressing: NAT Configuration Guide*.
TCP SYN Cookie	*Configuring Firewall TCP SYN Cookie* module.
VPN routing and forwarding (VRF)-aware firewall	*VRF-Aware Cisco IOS XE Firewall* module.
Virtual fragmentation reassembly (VFR)	*Virtual Fragmentation Reassembly* module.
Zone, default zone, and zone pair	*Zone-Based Policy Firewall* module.

Dual-Stack Firewalls

A dual-stack firewall is a firewall running IPv4 and IPv6 traffic at the same time. A dual-stack firewall can be configured in the following scenarios:

- One firewall zone running IPv4 traffic and another running IPv6 traffic.
- IPv4 and IPv6 coexist when deployed with stateful Network Address Translation 64 (NAT64). In this scenario, the traffic flows from IPv6 to IPv4 and vice versa.
- The same zone pair allows both IPv4 and IPv6 traffic.

Firewall Actions for IPv6 Header Fields

The firewall actions for IPv6 header fields (in the order they are available in the IPv6 header) are described in the following table:

Table 4: IPv6 Header Fields

IPv6 Header Field	IPv6 Header Field Description	Firewall Action
Version	Similar to the Version field in the IPv4 packet header, except that this field lists number 6 for IPv6, instead of number 4 for IPv4.	Must be IPv6.
Traffic Class	Similar to the Type of Service (ToS) field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.	Not inspected.
Flow Label	A new field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.	Not inspected.

IPv6 Header Field	IPv6 Header Field Description	Firewall Action
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.	The firewall uses this field on a limited basis to calculate the length of some of the Layer 4 protocols, such as ICMP and TCP.
Next Header Length	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header Length field determines the type of information that follows the basic IPv6 header. The type of information following the basic IPv6 header can be a transport-layer packet, for example, a TCP or a UDP packet, or an extension header.	The firewall must recognize this field to create a session.
Hop Limit	Similar to the Time-to-Live (TTL) field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of devices that an IPv6 packet can pass through before the packet is considered invalid. Each device decrements the Hop Limit value by one. Because the IPv6 header does not have a checksum, the device can decrement the value without recalculating the checksum.	Not inspected.

IPv6 Firewall Sessions

To perform stateful inspection of traffic, the firewall creates internal sessions for each traffic flow. The session information includes IP source and destination addresses, UDP or TCP source and destination ports or ICMP types, the Layer 4 protocol type (ICMP, TCP, or UDP), and VPN routing and forwarding (VRF) IDs. For an IPv6 firewall, the source and the destination addresses contain 128 bits of the IPv6 address.

The firewall creates a TCP session after receiving the first packet when the packet matches the configured policy. The firewall tracks the TCP sequence numbers and drops the TCP packets whose sequence numbers are not within the configured range. Sessions are removed when the TCP idle timer expires or when a Reset (RST) or Finish-Acknowledge (FIN-ACK) packet is received with the appropriate sequence numbers.

The firewall creates UDP sessions when the first UDP packet that matches the configured policy arrives and removes sessions when the UDP idle timer expires. The firewall does not create TCP or UDP sessions for IPv6 packets with multicast IPv6 or unknown IPv6 addresses.

Firewall Inspection of Fragmented Packets

The firewall supports the inspection of fragmented IPv6 packets. IP fragmentation is the process of breaking up a single IP datagram into multiple packets of smaller size. In IPv6, end nodes perform a path maximum transmission unit (MTU) discovery to determine the maximum size of the packet that is to be sent and generate IPv6 packets with the fragment extension header for packets larger than the MTU size.

The firewall inspects fragmented packets by using Virtual Fragmentation Reassembly (VFR). VFR examines the fragment extension header for out-of-sequence fragments and puts them in the correct order for inspection. When you enable the firewall on an interface by adding the interface to a zone, VFR is configured automatically on the same interface. If you explicitly disable VFR, the firewall only inspects the first fragments with Layer 4 headers and passes the rest of the fragments without inspection.

The fragment extension header appears in the following order of headers:

- IPv6 header
- Hop-by-hop options header
- Destination options header
- Routing header
- Fragment extension header

Cisco Express Forwarding checks IPv6 packets with fragment extension headers so that the firewall need not do further checks before processing the packets.

ICMPv6 Messages

IPv6 uses ICMPv6 to perform diagnostic functions, error reporting, and neighbor discovery. ICMPv6 messages are grouped into informational and error messages.

The firewall inspects only the following ICMPv6 messages:

- ECHO REQUEST
- ECHO REPLY
- DESTINATION UNREACHABLE
- PACKET TOO BIG
- PARAMETER PROBLEM
- TIME EXCEEDED

Note Neighbor discovery packets are passed and not inspected by the firewall.

Firewall Support of Stateful NAT64

The zone-based policy firewall supports Stateful NAT64. Stateful NAT64 translates IPv6 packets into IPv4 packets and vice versa. When both the firewall and Stateful NAT64 are configured on a router, the firewall uses IP addresses in an access control list (ACL) to filter packets. However, ACL does not support a mix of IPv4 and IPv6 addresses. Before the firewall and Stateful NAT64 can work together, you must use an IPv6 ACL and the IPv4 address must be embedded in the IPv6 ACL.

Note You cannot use VRF along with a firewall and a Stateful NAT64 configuration because Stateful NAT64 is not VRF-aware.

When a firewall class map uses an ACL, the ACL must use the real IP addresses on the host to configure packet flows. If only a source or a destination address is needed, either the IPv4 address or the IPv6 address is used in the class map ACL. Before the packet flow can be filtered based on both the source and destination addresses, the IPv6 address must be used and the IPv4 address must be embedded in the ACL. The ACL has to use IPv6 addresses to filter Stateful NAT64 packets.

Note Stateless NAT64 with firewall is not supported.

Port-to-Application Mapping

Port-to-application mapping (PAM) allows you to customize TCP or UDP port numbers for network services or applications. The firewall uses PAM to correlate TCP or UDP port numbers to specific network services or applications. By mapping port numbers to network services or applications, an administrator can force firewall inspection on custom configurations that are not defined by using well known ports. Use the **ip port-map** command to configure PAM.

High Availability and ISSU

The IPv6 firewall supports Intrabox HA. Firewall sessions are synchronized to the standby Embedded Services Processors (ESP) for a switchover. In Service Software Upgrade (ISSU) is also supported by the IPv6 firewall.

Pass Action for a Traffic Class

In a firewall, a traffic class identifies a set of packets based on its contents. You can define a class and apply an action to the identified traffic that reflects a policy. An action is a specific functionality that is associated with a traffic class. You can configure inspect, drop, and pass actions for a class.

The pass action passes the traffic from one zone to another. When the pass action is configured, the firewall does not inspect the traffic; it passes the traffic. In the IPv6 firewall, you must explicitly configure the pass action for the return traffic by defining a zone pair and a policy map with pass action.

The following example shows how to configure the pass action for policy maps, outside-to-inside-policy, and inside-to-outside-policy for IPv6 traffic:

```
policy-map type inspect outside-to-inside-policy
 class type inspect ipv6-class
  pass (Defines pass action for the ipv6-class from the outside to the inside)
!
 class class-default
!
policy-map type inspect inside-to-outside-policy
 class type inspect ipv4-class
  inspect (Defines inspect action for ipv4-class)
 class type inspect v6_class
  pass (Defines pass action for ipv6-class from the inside to the outside)
 class class-default
```

```
!
!
zone security inside
!
zone security outside
!
zone-pair security in-out source inside destination outside
 service-policy type inspect inside-to-outside-policy
!
zone-pair security out-in source outside destination inside
 service-policy type inspect outside-to-inside-policy
```

How to Configure Zone-Based Policy Firewall IPv6 Support

Configuring an IPv6 Firewall

The steps to configure an IPv4 firewall and an IPv6 firewall are the same. To configure an IPv6 firewall, you must configure the class map in such a way that only an IPv6 address family is matched.

The **match protocol** command applies to both IPv4 and IPv6 traffic and can be included in either an IPv4 policy or an IPv6 policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf-definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **sessions maximum** *sessions*
9. **exit**
10. **ipv6 unicast-routing**
11. **ip port-map** *appl-name* **port** *port-num* **list** *list-name*
12. **ipv6 access-list** *access-list-name*
13. **permit ipv6 any any**
14. **exit**
15. **class-map type inspect match-all** *class-map-name*
16. **match access-group name** *access-group-name*
17. **match protocol** *protocol-name*
18. **exit**
19. **policy-map type inspect** *policy-map-name*
20. **class type inspect** *class-map-name*
21. **inspect** [*parameter-map-name*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf-definition** *vrf-name* **Example:** `Device(config)# vrf-definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect ipv6-param-map`	Enables a global inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
Step 8	**sessions maximum** *sessions* **Example:** `Device(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 11	**ip port-map** *appl-name* **port** *port-num* **list** *list-name* **Example:**	Establishes a port to application mapping (PAM) by using the IPv6 access control list (ACL).

	Command or Action	Purpose
	`Device(config)# ip port-map ftp port 8090 list ipv6-acl`	
Step 12	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 13	**permit ipv6 any any** **Example:** `Device(config-ipv6-acl)# permit ipv6 any any`	Sets permit conditions for an IPv6 access list.
Step 14	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 15	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ipv6-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.
Step 16	**match access-group name** *access-group-name* **Example:** `Device(config-cmap)# match access-group name ipv6-acl`	Configures the match criteria for a class map on the basis of the specified ACL.
Step 17	**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 18	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 19	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ipv6-policy`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 20	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ipv6-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 21	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect ipv6-param-map`	Enables stateful packet inspection.

	Command or Action	Purpose
Step 22	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring Zones and Applying Zones to Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **ipv6 address** *ipv6-address/prefix-length*
12. **encapsulation dot1q** *vlan-id*
13. **zone-member security** *zone-name*
14. **end**
15. **show policy-map type inspect zone-pair sessions**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters 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 and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 5	**zone security** *zone-name* **Example:** `Device(config)# zone security z2`	Creates a security zone and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*] **Example:** `Device(config)# zone-pair security in-2-out source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy`	Attaches a policy map to a top-level policy map.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0.1`	Configures a subinterface and enters subinterface configuration mode.
Step 11	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-subif)# ipv6 address 2001:DB8:2222:7272::72/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface or a subinterface.
Step 12	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 2`	Sets the encapsulation method used by the interface.
Step 13	**zone-member security** *zone-name* **Example:** `Device(config-subif)# zone member security z1`	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. • 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 the

	Command or Action	Purpose
		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 14	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.
Step 15	**show policy-map type inspect zone-pair sessions** **Example:** `Device# show policy-map type inspect zone-pair sessions`	Displays the stateful packet inspection sessions created because a policy map is applied on a specified zone pair. • The output of this command displays both IPv4 and IPv6 firewall sessions.

Example

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv4 address and vice versa:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:32847=>(209.165.201.2:21) ftp SIS_OPEN
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [37:84]

        Half-open Sessions
         Session 110D930C [2001:DB8:1::104]:32848=>(209.165.201.2:21) ftp SIS_OPENING
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [0:0]
```

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv6 address:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:63=>[2001:DB8:2::102]:63 udp SIS_OPEN
```

```
Created 00:00:02, Last heard 00:00:01
Bytes sent (initiator:responder) [162:0]
```

Configuring an IPv6 Firewall and Stateful NAT64 Port Address Translation

The following task configures an IPv6 firewall with Stateful NAT64 dynamic port address translation (PAT).

A PAT configuration maps multiple IPv6 hosts to a pool of available IPv4 addresses on a first-come first-served basis. The dynamic PAT configuration directly helps conserve the scarce IPv4 address space while providing connectivity to the IPv4 Internet.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **no ip address**
6. **zone-member security** *zone-name*
7. **negotiation auto**
8. **ipv6 address** *ipv6-address/prefix-length*
9. **ipv6 enable**
10. **nat64 enable**
11. **exit**
12. **interface** *type number*
13. **ip address** *ip-address mask*
14. **zone member security** *zone-name*
15. **negotiation auto**
16. **nat64 enable**
17. **exit**
18. **ipv6 access-list** *access-list-name*
19. **permit ipv6 host** *source-ipv6-address* **host** *destination-ipv6-address*
20. **exit**
21. **ipv6 route** *ipv6-prefix/length interface-type interface-number*
22. **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*
23. **nat64 v4 pool** *pool-name start-ip-address end-ip-address*
24. **nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload**
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters 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	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 5	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone member security z1`	Attaches an interface to a security zone.
Step 7	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 8	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::2/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 9	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 10	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 on an interface.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 12	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 13	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.25 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 14	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security z2`	Attaches an interface to a security zone.
Step 15	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 16	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 on an interface.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 18	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-ipv4-pair`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 19	**permit ipv6 host** *source-ipv6-address* **host** *destination-ipv6-address* **Example:** `Device(config-ipv6-acl)# permit ipv6 host 2001:DB8:1::2 host 209.165:201.25`	Sets permit conditions for an IPv6 access list, a source IPv6 host address, and a destination IPv6 host address.
Step 20	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 21	**ipv6 route** *ipv6-prefix/length interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0`	Establishes static IPv6 routes.
Step 22	**ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address* **Example:** `Device(config)# ipv6 neighbor 2001:DB8:1::2/96 gigabitethernet 0/0/0 0000.29f1.4841`	Configures a static entry in the IPv6 neighbor discovery cache.

	Command or Action	Purpose
Step 23	**nat64 v4 pool** *pool-name start-ip-address end-ip-address* **Example:** `Device(config)# nat64 v4 pool pool1 209.165.201.25 209.165.201.125`	Defines a Stateful NAT64 IPv4 address pool.
Step 24	**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload** **Example:** `Device(config)# nat64 v6v4 list nat64-ipv6-any pool pool1 overload`	Enables NAT64 PAT or overload address translation.
Step 25	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Zone-Based Policy Firewall IPv6 Support

Example: Configuring an IPv6 Firewall

```
Device# configure terminal
Device(config)# vrf-definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# parameter-map type inspect ipv6-param-map
Device(config-profile)# sessions maximum 10000
Device(config-profile)# exit
Device(config)# ipv6 unicast-routing
Device(config)# ip port-map ftp port 8090 list ipv6-acl
Device(config)# ipv6 access-list ipv6-acl
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# class-map type inspect match-all ipv6-class
Device(config-cmap)# match access-group name ipv6-acl
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect ipv6-policy
Device(config-pmap)# class type inspect ipv6-class
Device(config-pmap-c)# inspect ipv6-param-map
Device(config-pmap-c)# end
```

Example: Configuring Zones and Applying Zones to Interfaces

```
Device# configure terminal
Device(config)# zone security z1
```

```
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security in-to-out source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0.1
Device(config-if)# ipv6 address 2001:DB8:2222:7272::72/64
Device(config-if)# encapsulation dot1q 2
Device(config-if)# zone member security z1
Device(config-if)# end
```

Example: Configuring an IPv6 Firewall and Stateful NAT64 Port Address Translation

```
configure terminal
 ipv6 unicast-routing
 interface gigabitethernet 0/0/0
  no ip address
  zone member security z1
  negotiation auto
  ipv6 address 2001:DB8:1::2/96
  ipv6 enable
  nat64 enable
!
 interface gigabitethernet 0/0/1
  ip address 209.165.201.25 255.255.255.0
  zone member security z2
  negotiation auto
  nat64 enable
!
 ipv6 access-list ipv6-ipv4-pair
  permit ipv6 host 2001:DB8:1::2 host 209.165:201.25
!
 ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0
 ipv6 neighbor 2001:DB8:1::2/96 gigabitethernet 0/0/0 0000.29f1.4841
 nat64 v4 pool pool1 209.165.201.25 209.165.201.125
 nat64 v6v4 list nat64-ipv6-any pool pool1 overload
```

Additional References for Zone-Based Policy Firewall IPv6 Support

Related Documents

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

Related Topic	Document Title
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
Stateful NAT64	Stateful Network Address Translation 64

Standards and RFCs

Standard/RFC	Title
RFC 2460	*Internet Protocol, Version 6 (IPv6) Specification*
RFC 2473	*Generic Packet Tunneling in IPv6 Specification*

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

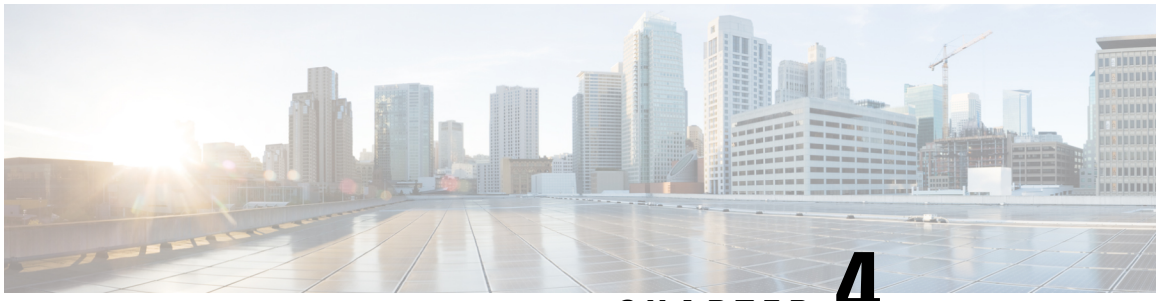
Feature Information for Zone-Based Policy Firewall IPv6 Support

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 5: Feature Information for Zone-Based Policy Firewall IPv6 Support

Feature Name	Releases	Feature Information
Zone-Based Policy Firewall IPv6 Support	Cisco IOS XE Release 3.6S	The Zone-Based Policy firewall supports the inspection of IPv6 packets. The following commands were introduced or modified: **ip port-map** and **show policy-map type inspect zone-pair**.



CHAPTER 4

VRF-Aware Cisco IOS XE Firewall

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

The VRF-Aware Cisco IOS XE Firewall supports VRF-lite (also known as Multi-VRF CE) and Application Inspection and Control (AIC) for various protocols.

The VRF-aware firewall supports VRF-lite (also known as Multi-VRF CE) and Application Inspection and Control (AIC) for various protocols.

Note Cisco IOS XE Releases do not support Context-Based Access Control (CBAC) 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.

Prerequisites for VRF-Aware Cisco IOS XE Firewall

- Understand Cisco IOS XE firewalls.
- Configure VRFs.

Restrictions for VRF-Aware Cisco IOS XE Firewall

- If two VPN networks have overlapping addresses, VRF-aware Network Address Translation (NAT) is required for them to support VRF-aware firewalls. NAT does not support inter-VRF routing. You can use the VRF-aware software infrastructure (VASI) for the inter-VRF routing functionality.
- You cannot apply per-VRF firewall policies if crypto tunnels that belong to multiple VPNs terminate on a single interface.
- Site-Site crypto maps on VASI interfaces are not supported on the following platforms:
  - Cisco 1000 Series Integrated Services Routers
  - Cisco 4000 Series Integrated Services Routers
  - Cisco 1000v Cloud Services Routers
- The same zone cannot be applied to interfaces that are configured on different VRFs.

Information About VRF-Aware Cisco IOS XE Firewall

VRF-Aware Cisco IOS XE Firewall

A VRF-aware firewall inspects IP packets that are sent or received within a VRF. VRF allows multiple instances of routing tables to coexist within a single router. This allows VPN segregation and the ability to have independent overlapping of IP address spaces. VRF allows traffic from the customers of one service provider to be isolated from another. The Cisco IOS XE VRF support splits the router into multiple routing domains, with each routing domain consisting of its own set of interfaces and routing and forwarding tables. Each routing domain is referenced by a unique identifier called the table ID. The global routing domain and the default routing domain (that is not associated with any VRF) is addressed with the table ID, zero. VRF supports overlapping of IP address space, thereby allowing the traffic from nonintersecting VRFs to have the same IP address.

The VRF-Aware Cisco IOS XE Firewall provides the following benefits:

- Scalable deployment—Scales to meet any network's bandwidth and performance requirements.
- VPN support—Provides a complete VPN solution based on Cisco IOS XE IPsec and other software-based technologies, including Layer 2 Tunneling Protocol (L2TP) tunneling, and quality of service (QoS).
- AIC support—Provides policy maps for the Internet Message Access Protocol (IMAP), Post Office Protocol 3 (POP3), Simple Mail Transfer Protocol (SMTP), and Sun Remote Procedure Call (SUN RPC)

- Allows users to configure a per-VRF firewall. The firewall inspects IP packets that are sent and received within a VRF. The firewall also inspects traffic between two different VRFs (intersecting VRFs).
- Allows SPs to deploy the firewall on the provider edge (PE) router.
- Supports overlapping IP address space, thereby allowing traffic from nonintersecting VRFs to have the same IP address.
- Supports 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) that are allocated to various VPN customers.
- Generates high-speed logging (HSL) messages that contain the VRF ID; however these messages are collected by a single collector.

The VRF-aware firewall allows you to limit the number of firewall sessions. If the firewall sessions are not limited, it would be difficult for VRFs to share router resources because one VRF may consume a maximum amount of resources, leaving few resources for other VRFs and thereby causing the denial of service to other VRFs.

Note On the Cisco ASR 1000 Series Aggregation Services Routers the firewall supports a maximum of 4000 VRFs.

Address Space Overlap

A VRF splits the device into multiple routing domains. Each of these routing domains contain their own set of interfaces and routing tables. A routing table is referenced by using a per-VRF unique table ID. Zero is the default global routing table ID that is not associated with a VPN routing and forwarding (VRF).

Nonintersecting VRFs are allowed to have overlapping address spaces (that is, the IP address of one VRF may be contained in others).

VRF

VPN routing and forwarding (VRF) allows multiple instances of routing tables to coexit within a single device. A VRF contains a template of a VRF table in a provider edge (PE) device.

The overlapping addresses, usually resulting from the use of private IP addresses in customer networks, are one of the major obstacles to the successful deployment of a peer-to-peer (P2P) VPN implementation. You can use the Multiprotocol Label Switching (MPLS) VPN technology to overcome the overlapping addresses issue.

Each VPN has its own routing and forwarding table in the device so that any customer or site that belongs to a VPN is provided access only to the set of routes contained within that table. Any PE device 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 devices in the service provider (SP) network. Effectively, a number of virtual devices are created in a single physical device.

VRF-Lite

The VRF-Lite Aware Firewall feature, also called the VRF without MPLS-aware firewall, allows a firewall zone to be applied to non-MPLS-enabled VPN routing and forwarding (VRF) interfaces.

The VRF-Lite Aware Firewall feature enables a service provider (SP) to support two or more VPNs, in which IP addresses can be overlapped among 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.

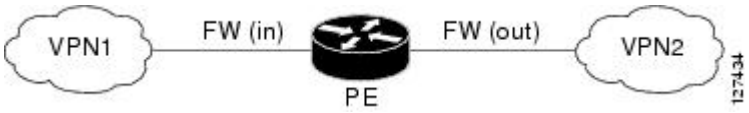
Note All VRF-lite interfaces must be Layer 3 interfaces.

VRF-lite includes the following devices:

- Customer edge (CE) devices provide customers access to the SP network over a data link. The CE device advertises the site's local routes to the provider edge (PE) device and learns about the remote VPN routes from the PE device.
- PE devices exchange routing information with CE devices by using static routing or a routing protocol such as Border Gateway Protocol (BGP), Routing Information Protocol Version 1 (RIPv1), or RIPv2.
- PE devices (or core devices) are any devices in the SP network that are not attached to CE devices.
- A PE device is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE device to maintain all the SP VPN routes. Each PE device maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE device can be associated with a single VRF, if all of these sites are part of the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CE devices, a PE device exchanges VPN routing information with other PE devices by using internal BGP (iBPG).

With VRF-lite, multiple customers can share one CE device, and only one physical link is used between the CE device and the PE device. The shared CE device maintains a separate VRF table for each customer, and switches or routes packets for each customer based on its own routing table. VRF-lite extends the limited PE device 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.

Figure 6: Firewall in a VRF-to-VRF Scenario



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 device, 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 possess Network Information Center (NIC)-registered IP addresses must acquire them. NAT eliminates the concern of NIC-registered IP addresses by dynamically mapping thousands of hidden internal addresses to a range of easy-to-get addresses.

A NAT system makes it difficult for an attacker to determine the following:

- Number of systems running on a network.
- Type of machines and operating systems running on the network.
- Network topology and arrangement.

NAT integration with Multiprotocol Label Switching (MPLS) VPNs allows multiple MPLS VPNs to be configured on a single device to work together. NAT can differentiate the MPLS VPNs from which it receives the IP traffic, even if all MPLS VPNs use 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.

To provide value-added services, such as, Internet connectivity, domain name servers (DNS), and VoIP service to customers, MPLS service providers must use NAT. NAT helps MPLS VPN customers to use overlapped IP addresses in their network.

NAT can be implemented on a customer edge (CE) device or on a provider edge (PE) device. The NAT integration with MPLS VPNs feature enables the implementation of NAT on a PE device in an MPLS cloud.

VRF-Aware ALG

An application-layer gateway (ALG) is an application that translates the IP address information inside the payload of an application packet. The ALGs identify the address information in the packet payload that needs to be overwritten by NAT and supply the address information to NAT and firewall to create subordinate flows or doors to allow data to flow properly (an example of data flow is FTP data flow. Doors are transient structures that allow incoming traffic that matches a specific criterion. A door is created when there is not enough information to create a complete NAT session entry. A door contains information about the source and destination IP address and the destination port. However, it does not have information about the source port. When media data arrives, the source port information is known and the door is promoted to a real NAT session.

VRF-Aware IPsec

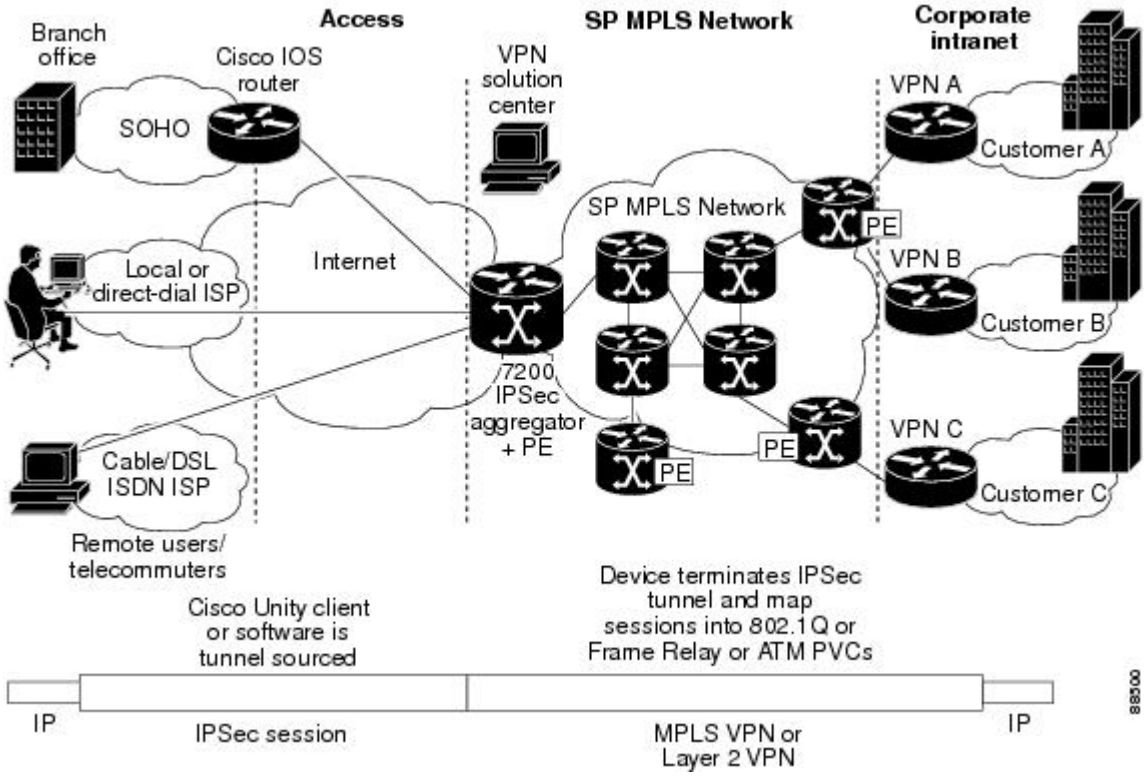
The VRF-Aware IPsec feature maps an IPsec tunnel to a Multiprotocol Label Switching (MPLS) VPN. Using the VRF-Aware IPsec feature, you can map IPsec tunnels to VPN routing and forwarding (VRF) instances using a single public-facing IP address.

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 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 following figure illustrates a scenario showing IPsec to MPLS and Layer 2 VPNs.

Figure 7: IPsec-to-MPLS and Layer 2 VPNs



VRF-Aware Software Infrastructure

The VRF-Aware Software Infrastructure (VASI) allows you to apply services such as access control lists (ACLs), NAT, policing, and zone-based firewalls to traffic that is flowing across two different VRF instances. The VASI interfaces support redundancy of the Route Processor (RP) and Forwarding Processor (FP). This feature supports IPv4 and IPv6 unicast traffic on VASI interfaces.

The primary use of VASI is to allow better isolation of VRFs. The VASI allows for per-VRF-specific features to be applied to the VASI interface without any impact to other VRFs that may share a common interface (for example, all VRFs may share the same interface to the Internet). For the firewall, this feature allows zones to be applied to the VASI.

VASI is implemented by using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF. The VASI virtual interface is the next hop interface for any packet that needs to be switched between these two VRFs. VASI interfaces provide the framework necessary to support NAT between two VRFs.

Each interface pair is associated with two different VRF instances. The two virtual interfaces, called vasileft and vasiright, in a pair are logically wired back-to-back and are completely symmetrical. Each interface has an index. The association of the pairing is done automatically based on the two interface indexes such that vasileft automatically gets paired to vasiright. You can configure either static routing or dynamic routing with BGP, Enhanced Interior Gateway Routing Protocol (EIGRP), or Open Shortest Path First (OSPF). BGP dynamic routing protocol restrictions and configuration are valid for BGP routing configurations between VASI interfaces. For more information on VASI, see the " *Configuring the VRF-Aware Software Infrastructure* " feature.

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:

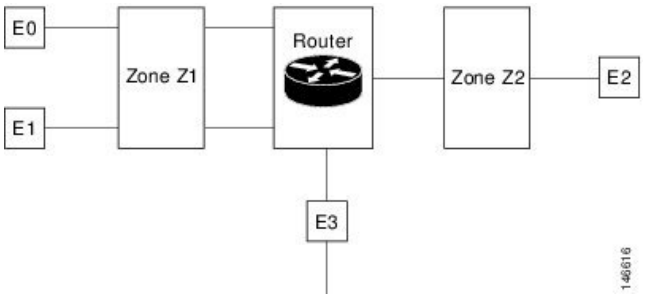
- Traffic from a zone interface to a nonzone interface or from a nonzone interface to a zone interface is always dropped; unless default zones are enabled (default zone is a nonzone interface).
- Traffic between two zone interfaces is inspected if there is a zone pair relationship for each zone and if 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, pass or drop the traffic between the two zones.
- 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.
- For traffic to flow among all interfaces in a device, these interfaces must be members of one security zone or another. It is not necessary for all device interfaces to be members of security zones.

- All interfaces associated with a zone must be contained in the same VRF (Virtual Routing Forwarding).

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 8: Security Zone Restrictions



The following situations exist:

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

Note On the Cisco ASR 1000 Series Aggregation Services Routers the firewall supports a maximum of 4000 zones.

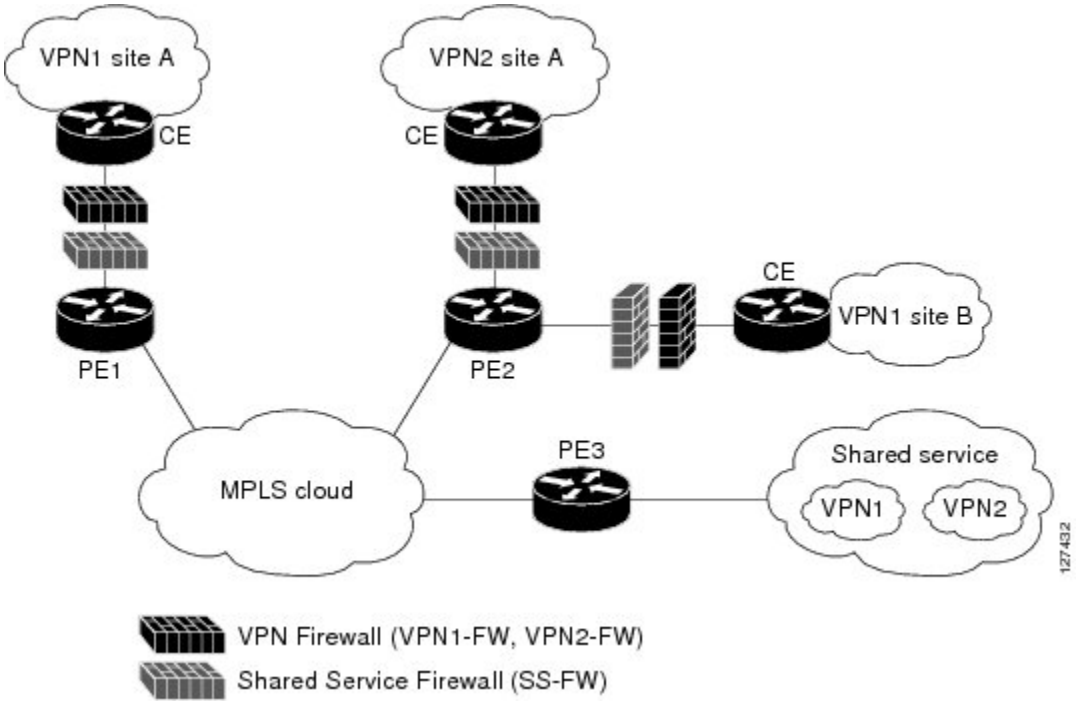
VRF-Aware Cisco Firewall Deployment

A firewall can be deployed at many points within the network to protect VPN sites from shared service (or the Internet) and vice versa. This section describes the following firewall deployment scenarios:

Distributed Network Inclusion of VRF-Aware Cisco Firewall

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

Figure 9: Distributed Network



In this example, VPN1 has two sites, Site A and Site B, that span across the Multiprotocol Label Switching (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 has a VLAN segment in the shared service that is connected to the corresponding VLAN subinterface on PE3.

Each of the VPNs (VPN1 and VPN2) has two firewall rules—one to protect the VPN site from the shared service and another to protect the shared service from the VPN site. The firewall that protects the VPN site from the shared service is called the VPN firewall, and the firewall that protects the shared service from the VPN site is called the shared service firewall. Both firewall rules are applied on the VPN routing and forwarding (VRF) interface of each ingress provider edge (PE) device that is connected to the VPN site. The VPN firewall rule is applied in the ingress direction, because the VRF interface is ingress to the VPN site; and the shared service firewall rule is applied in the egress direction, because the VRF interface is egress to the shared service.

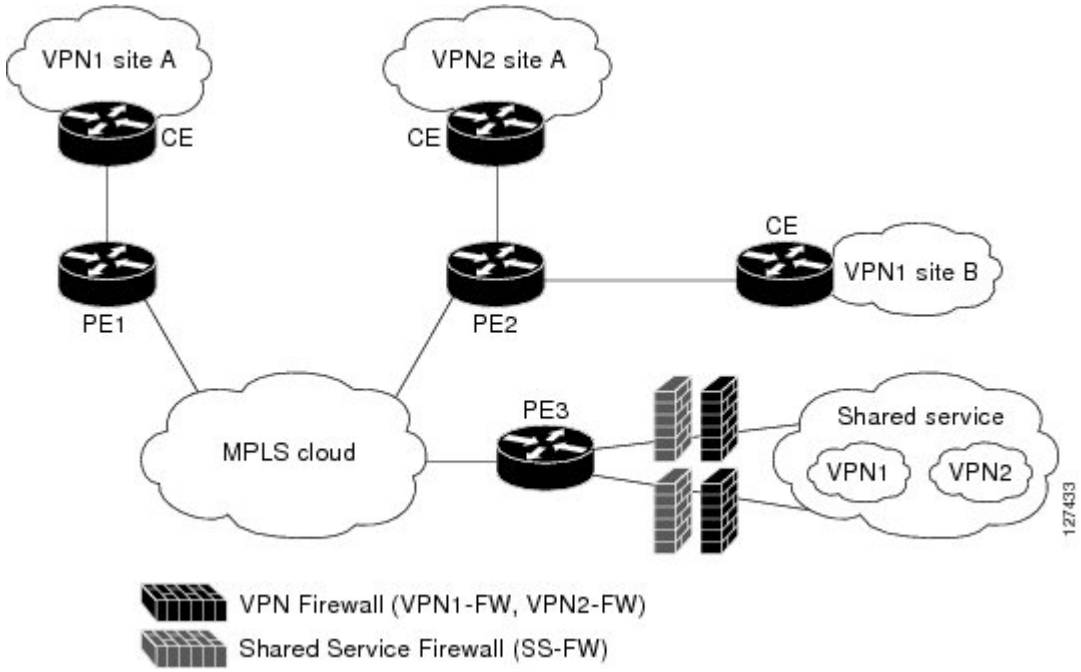
The benefits of using a distributed network are as follows:

- Because the firewall deployment is distributed across a Multiprotocol Label Switching (MPLS) cloud, the firewall processing load is distributed to all ingress PE devices.
- The shared service is protected from VPN sites at the ingress PE device, and hence malicious packets from VPN sites are filtered at the ingress PE device before they enter the MPLS cloud.
- VPN firewall features can be deployed in the ingress direction.

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

The following figure illustrates a hub-and-spoke network where firewalls for all VPN sites are applied on the egress PE device, PE3, which is connected to the shared service.

Figure 10: Hub-and-Spoke Network



Typically, each VPN has a VLAN and/or a VPN routing and forwarding (VRF) subinterface that is connected to the shared service. When a packet arrives at a Multiprotocol Label Switching (MPLS) interface, MPLS routes the packet to the corresponding subinterface that is connected to the shared service. Firewall policies on each VPN are applied on the corresponding subinterface (VRF interface) as shown in the above figure. The VPN firewall rule is applied in the egress direction because the subinterface is egress to the VPN site. And the shared service firewall rule is applied in the ingress direction because the subinterface is ingress to the shared service.

The benefits of a hub-and-spoke network are as follows:

- Because the firewall deployment is centralized to the egress provider edge (PE) device (PE3), deploying and managing the firewall is easy.
- The shared service firewall feature can be applied in the ingress direction.
- The VPN site is protected from the shared service at the egress PE device, and hence malicious packets from the shared service are filtered at the PE device before they enter the MPLS cloud.

How to Configure VRF-Aware Cisco IOS XE Firewall

Defining VRFs, Class Maps, and Policy Maps

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **exit**
8. **class-map type inspect match-any** *class-map-name*
9. **match protocol tcp**
10. **match protocol h323**
11. **exit**
12. **policy-map type inspect** *policy-map-name*
13. **class type inspect** *class-map-name*
14. **inspect** [*parameter-map-name*]
15. **exit**
16. **class class-default**
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Defines a VRF instance and to enter VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 10:1`	Specifies a route distinguisher (RD) for a VRF instance.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Router(config-vrf)# route-target export 10:1`	Creates a route-target extended community for a VRF instance and exports routing information to the target VPN extended community.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 10:1`	Creates a route-target extended community for a VRF instance and imports routing information to the target VPN extended community.
Step 7	**exit** **Example:**	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Router(config-vrf)# exit`	
Step 8	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any class-map1`	Creates a Layer 3 and Layer 4 (application-specific) inspect type class map and enters class-map configuration mode.
Step 9	**match protocol tcp** **Example:** `Router(config-cmap)# match protocol tcp`	Configures the match criterion for a class map on the basis of the specified protocol.
Step 10	**match protocol h323** **Example:** `Router(config-cmap)# match protocol h323`	Configures the match criterion for a class map on the basis of the specified protocol.
Step 11	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode and enters global configuration mode.
Step 12	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect global-vpn1-pmap`	Creates a Layer 3 and Layer 4 (protocol-specific) inspect type policy map and enters policy-map configuration mode.
Step 13	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect class-map1`	Specifies the traffic (class) on which an action is to be performed and enters policy-map-class configuration mode.
Step 14	**inspect** [*parameter-map-name*] **Example:** `Router(config-pmap-c)# inspect class-map1`	Enables Cisco IOS XE stateful packet inspection.
Step 15	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map-class configuration mode and enters policy-map configuration mode.
Step 16	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies the default class so that you can configure or modify its policy. • The class-default class is defined by default. Configure the **class class-default** command to change the default drop attribute that is associated with the class-default.
Step 17	**end** **Example:** `Router(config-pmap)# end`	Exits policy-map configuration mode and enters global configuration mode.

Defining Zones and Zone Pairs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *security-zone-name*
4. **exit**
5. **zone security** *security-zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
8. **service-policy type inspect** *policy-map-name*
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	**zone security** *security-zone-name* **Example:** `Router(config)# zone security vpn1-zone`	Creates a security zone 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.
Step 5	**zone security** *security-zone-name* **Example:** `Router(config)# zone security global-zone`	Creates a security zone 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-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:**	Creates a zone pair and enters security zone-pair configuration mode. • *zone-pair-name*--Name of the zone being attached to an interface.

	Command or Action	Purpose
	`Router(config)# zone-pair security vpn1-global-zone-pair source vpn1-zone destination global-zone`	• **source** *source-zone*--Specifies the name of the router from which traffic is originating. • **destination** *destination-zone*--Specifies the name of the router to which traffic is bound.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect global-vpn1-pmap`	Attaches a Layer 7 policy map to a top-level policy map.
Step 9	**end** **Example:** `Router(config-sec-zone-pair)# end`	Exits zone-pair configuration mode and enters privileged EXEC mode.

Applying Zones to Interfaces and Defining Routes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *name*
5. **ip address** *ip-address mask*
6. **zone-member security** *zone-name*
7. **negotiation auto**
8. **exit**
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **zone-member security** *zone-name*
12. **negotiation auto**
13. **exit**
14. **ip route vrf** *vrf-name destination-ip-address destination-prefix interface-type number* [**global**]
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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip vrf forwarding** *name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 6	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security  vpn1-zone`	Attaches an interface to a security zone.
Step 7	**negotiation auto** **Example:** `Router(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/1/1`	Configures an interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.111.111.111 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 11	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security global-zone`	Attaches an interface to a security zone.
Step 12	**negotiation auto** **Example:** `Router(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 14	**ip route vrf** *vrf-name destination-ip-address destination-prefix interface-type number* [**global**] **Example:** `Router(config)# ip route vrf vpn1 10.111.111.0 255.255.255.0 gigabitethernet 1/1/1 global`	Establishes static routes for a VRF instance.
Step 15	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for VRF-Aware Cisco IOS XE Firewall

Example: Defining VRFs, Class Maps, and Policy Maps

```
Router# configure terminal
Router(config)# ip vrf vrf1
Router(config-vrf)# rd 10:1
Router(config-vrf)# route-target export 10:1
Router(config-vrf)# route-target import 10:1
Router(config-vrf)# exit
Router(config)# class-map type inspect match-any class-map1
Router(config-cmap)# match protocol tcp
Router(config-cmap)# match protocol h323
Router(config-cmap)# exit
Router(config)# policy-map type inspect global-vpn1-pmap
Router(config-pmap)# class type inspect match-acl-111
Router(config-pmap-c)# inspect match-acl-111
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap)# end
```

Example: Defining Policy Maps, Zones, and Zone Pairs

```
Router# configure terminal
Router(config)# zone security vpn1-zone
Router(config-sec-zone)# exit
Router(config)# zone security global-zone
Router(config-sec-zone)# exit
Router(config)# zone-pair security vpn1-global-zone-pair source vpn1-zone destination
global-zone
Router(config-sec-zone-pair)# service-policy type inspect vpn1-global-pmap
Router(config-sec-zone-pair)# end
```

Example: Applying Zones to Interfaces and Defining Routes

```
Router# configure terminal
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# zone-member security vpn1-zone
Router(config-if)# negotiation auto
Router(config-if)# exit
Router(config)# interface gigabitethernet 1/1/1
Router(config-if)# ip address 10.111.111.111 255.255.255.0
Router(config-if)# zone-member security global-zone
Router(config-if)# negotiation auto
Router(config-if)# exit
Router(config)# ip route vrf vpn1 10.111.111.0 255.255.255.0 gigabitethernet 1/1/1 global
Router(config)# end
```

Additional References for VRF-Aware Cisco IOS XE Firewall

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
NAT	Configuring Network Address Translation: Getting Started
MPLS VPN	*onfiguring a Basic MPLS VPN*
Zone-based Policy Firewall	*Zone-based Policy Firewall*

Technical Assistance

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

Feature Information for VRF-Aware Cisco IOS XE 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 6: Feature Information for VRF-Aware Cisco IOS XE Firewall

Feature Name	Releases	Feature Information
VRF-Aware Cisco IOS XE Firewall	Cisco IOS XE Release 2.5	The VRF-Aware Cisco IOS XE Firewall feature applies the Cisco IOS XE Firewall functionality to VRF interfaces when the firewall is configured on an SP or large enterprise edge router.
Firewall--VRF-Aware ALG Support	Cisco IOS XE Release 2.5	The Firewall--VRF-Aware ALG Support feature allows ALG to extract the correct IP address and VRF ID from cached information when creating ALG tokens that require correct IP address VRF ID pairs.

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.

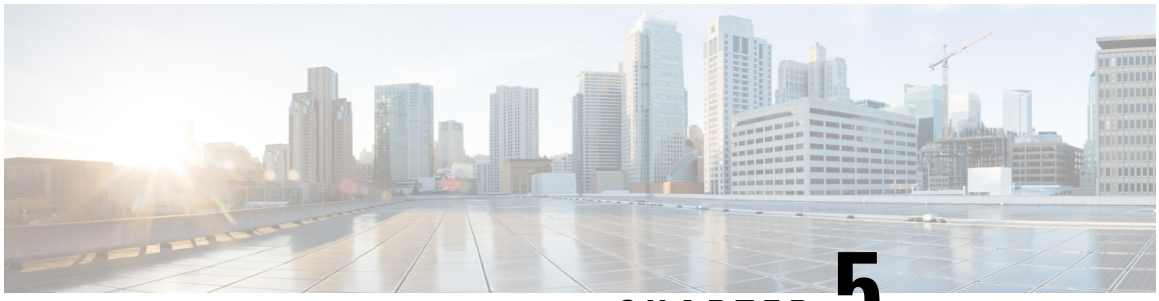
HELO --Command that starts the SMTP capability negotiation. This command identifies the sender (client) connecting to the remote SMTP server by its fully qualified DNS hostname.

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 5

Layer 2 Transparent Firewalls

A Layer 2 transparent firewall operates on bridged packets and is enabled on a pair of locally-switched Ethernet ports. Embedded IP packets forwarded through these ports are inspected similar to normal IP packets in a routing network. The zone-based firewall or Layer 3 firewall configuration can be applied to Layer 2 interfaces for the transparent firewall configuration.

This module provides an overview of the Layer 2 Transparent 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 Layer 2 Transparent Firewalls Support

- Address Resolution Protocol (ARP) inspection is not supported.
- Layer 2 forwarding technologies such as bridge domain, bridge domain interfaces (BDI), Overlay Transport Virtualization (OTV), X-Connect, Virtual Private LAN Services (VPLS), VxLAN, and non-IP flows, are not supported.
- Only normal IP or simple VLAN is supported on Ethernet frames. The transparent firewall generates TCP reset (RST) packets and sends these packets in supported Ethernet frame.
- TCP RST is not supported after intrabox high availability switchover.
- Virtual TCP (vTCP) is not supported.

- Network Address Translation (NAT), Box-to-Box (B2B) high availability, Multiprotocol Label Switching (MPLS), Virtual Routing and Forwarding (VRF) instances, VRF-Aware Software Infrastructure (VASI), Locator-ID Separation Protocol (LISP) are not supported in the Layer 2 switch path.
- Non IP packet flows like Ethernet Operation, Administration, and Maintenance (OAM), Connectivity Fault Management (CFM) is not supported.
- Layer 2-based access control lists (ACLs) are not supported in the transparent firewall class map.

Information About Layer 2 Transparent Firewalls

Layer 2 Transparent Firewall Support

A traditional zone-based firewall acts like a Layer 3 node in a network, and inspects the IP traffic that passes through the node. The traditional firewall is a routed hop and acts as a default gateway for hosts that connect to one of its screened subnets. However, to place this Layer 3 firewall in an existing network requires the network to be re-subnetted, which is time and resource-intensive. The Layer 2 transparent firewall is transparent to the network and does not require Layer 3 separation between segments. A transparent firewall acts like a "bump in the wire" or a "stealth firewall," and is not seen as a router hop to connected devices. Because the firewall is not a routed hop, you can easily introduce a transparent firewall into an existing network; IP readdressing is unnecessary. The transparent firewall operates on bridged packets and the Layer 3 firewall operates on routed packets.

A transparent firewall is enabled on a pair of locally-switched Ethernet ports. Embedded IP packets forwarded through these ports are inspected similar to normal IP packets in a routing network. The transparent firewall only inspects IP packets.

A transparent firewall session is created by using IP Layer 3 and Layer 4 headers that contain 5-tuple information (5-tuple information are source and destination IP addresses, source and destination ports, and the protocol). The transparent firewall supports only Ethernet as a Layer 2 protocol, and supports both IPv4 and IPv6 addresses.

The zone-based firewall or Layer 3 firewall configuration can be applied to Layer 2 interfaces for the transparent firewall configuration. Both Layer 3 firewall and Layer 2 transparent firewall can coexist on a device.

The transparent firewall supports IP (Internet Control Message Protocol [ICMP], TCP, and UDP) inspection with the following topologies:

- Between two GigabitEthernet interfaces.
- Between a GigabitEthernet interface and a GigabitEthernet subinterface.
- Between two GigabitEthernet subinterfaces

The transparent firewall passes the following packets without a policy attached to them:

- Address Resolution Protocol (ARP)
- Multicast packets: Routing Information Protocol (RIP), Open Shortest Path First (OSPF), OSPF Version 3 (OSPFv3), Enhanced Interior Gateway Routing Protocol (EIGRP) IPv4 and IPv6 packets, Intermediate System-to-Intermediate System (ISIS) IPv4 and IPv6 packets
- Protocol-Independent Multicast (PIM) IPv4 and IPv6 packets

- Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), and Gateway Load Balancing Protocol (GLBP)
- Internet Group Management Protocol (IGMP), and Multicast Listener Discovery (MLD)

How to Configure Layer 2 Transparent Firewalls

You can configure a Layer 2 transparent firewall using the same configuration as the zone-based firewalls. For more information, see the "Zone-Based Firewalls" module.

Configuration Examples for Layer 2 Transparent Firewalls

Example: Configuring a Layer 2 Transparent Firewall

The following example shows how to configure a Layer 2 transparent firewall with TCP and UDP inspection:

- Defines class maps.
- Defines policy maps.
- Defines zones and zone pairs.
- Attaches interfaces GigabitEthernet 0/0/0 and GigabitEthernet 0/0/1 to firewall zones.
- Enables local switching by connecting GigabitEthernet 0/0/0 with GigabitEthernet 0/0/1.

```
!Class map configuration
Device# configure terminal
Device(config)# class-map typ inspect match-any lan-wan-inspect-tcp
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol udp
Device(config-cmap)# match protocol icmp
Device(config-cmap)# exit
Device(config-cmap)# exit
Device(config)# class-map type inspect match-any wan-lan-inspect-udp
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol udp
Device(config-cmap)# match protocol icmp
Device(config-cmap)# exit

Device(config-cmap)# exit

!Policy map configuration
Device(config)# policy-map type inspect policy-wan-lan
Device(config-pmap)# class type inspect lan-wan-inspect-tcp
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap)# class type inspect wan-lan-inspect-udp
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
```

```
Device(config-pmap-c)# exit
Device(config-pmap)# exit

!Zones and zone pair configuration
Device(config)# zone security lan
Device(config-sec-zone)# exit
Device(config)# zone security wan
Device(config-sec-zone)# exit
Device(config)# zone-pair security lan2wan source lan destination wan
Device(config-sec-zone-pair)# service-policy type inspect policy-lan-wan
Device(config-sec-zone-pair)# exit
Device(config)# zone-pair security wan2lan source wan destination lan
Device(config-sec-zone-pair)# service-policy type inspect policy-wan-lan
Device(config-sec-zone-pair)# exit

! Interface configuration
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# no ip address
Device(config-if)# zone-member security lan
Device(config-if)# exit
Device(config)# interface gigiabitethernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# zone-member security wan
Device(config-if)# exit

!Local switching configuration
Device(config)# connect l2fw-conn gigabitethernet 0/0/0 gigabitethernet 0/0/1
Device(config)# end
```

Additional References for Layer 2 Transparent 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
Zone-based firewalls	"Zone-Based Policy Firewalls" module in the *Zone-Based Policy Firewalls*, Configuration Guide.

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

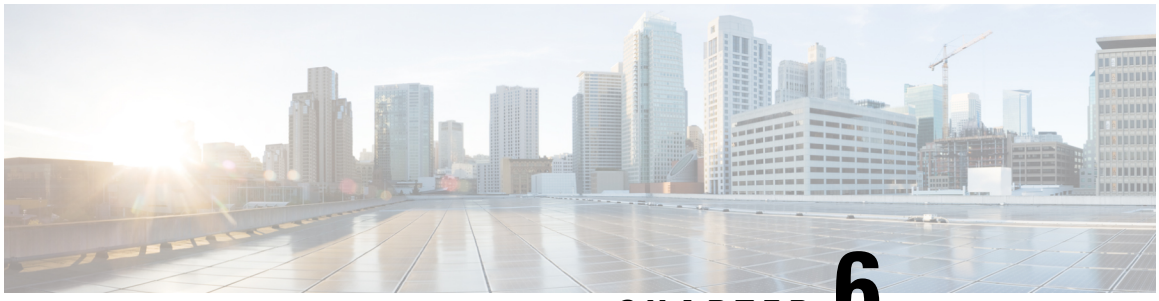
Feature Information for Layer 2 Transparent Firewalls

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 7: Feature Information for Layer 2 Transparent Firewalls

Feature Name	Releases	Feature Information
Layer 2 Transparent Firewalls	Cisco IOS XE 3.15S	A Layer 2 transparent firewall operates on bridged packets and is enabled on a pair of locally-switched Ethernet ports. Embedded IP packets forwarded through these ports are inspected similar to normal IP packets in a routing network. The zone-based firewall or Layer 3 firewall configuration can be applied to Layer 2 interfaces for the transparent firewall configuration. This feature is supported on Cisco ASR 1000 Series Aggregation Services Routers, and Cisco Cloud Services Router 1000V Series. No commands were introduced or updated for this feature.



CHAPTER 6

Nested Class Map Support for Zone-Based Policy Firewall

The Nested Class Map Support for Zone-Based Policy Firewall feature provides the Cisco IOS XE firewall the functionality to configure multiple traffic classes (which are also called nested class maps or hierarchical class maps) as a single traffic class. When packets meet more than one match criterion, you can configure multiple class maps that can be associated with a single traffic policy. The Cisco IOS XE firewall supports up to three levels of class map hierarchy.

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 Nested Class Map Support for Zone-Based Policy Firewall

Before configuring nested class maps, you should be familiar with the modular Quality of Service (QoS) CLI (MQC).

Information About Nested Class Map Support for Zone-Based Policy Firewall

Nested Class Maps

In Cisco IOS XE Release 3.5S and later releases, you can configure multiple traffic classes (which are also called nested class maps or hierarchical class maps) as a single traffic class. When packets meet more than one match criterion, you can configure multiple class maps that can be associated with a single traffic policy. The nesting of class maps can be achieved by configuring the **match class-map** command. The only method of combining the match-any and match-all characteristics within a single traffic class is by using the **class-map** command.

match-all and match-any Keywords of the class-map Command

To create a traffic class, you must configure the **class-map** command with the **match-all** and **match-any** keywords. You need to specify the **match-all** and **match-any** keywords only if more than one match criterion is configured in the traffic class. The following rules apply to the **match-all** and **match-any** keywords:

- Use the **match-all** keyword when all match criteria in the traffic class must be met to place a packet in the specified traffic class.
- Use the **match-any** keyword when only one of the match criterion in the traffic class must be met to place a packet in the specified traffic class.
- If you do not specify the **match-all** keyword or the **match-any** keyword, the traffic class behaves in a manner that is consistent with the **match-all** keyword.

Your zone-based policy firewall configuration supports nested class maps if the following criteria are met:

- Individual class maps in a hierarchy include multiple **match class-map** command references.
- Individual class maps in a hierarchy include match rules other than the **match class-map** command.

How to Configure Nested Class Map Support for Zone-Based Policy Firewall

Configuring a Two-Layer Nested Class Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**

6. **class-map match-any** *class-map-name*
7. **match protocol** *protocol-name*
8. **exit**
9. **class-map match-any** *class-map-name*
10. **match class-map** *class-map-name*
11. **match class-map** *class-map-name*
12. **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 match-any** *class-map-name* **Example:** `Router(config)# class-map match-any child1`	Creates a Layer 3 or Layer 4 class map and enters class map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol tcp`	Configures the match criteria for a class map on the basis of a specified protocol.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits class map configuration mode and enters global configuration mode.
Step 6	**class-map match-any** *class-map-name* **Example:** `Router(config)# class-map match-any child2`	Creates a Layer 3 or Layer 4 class map and enters class map configuration mode.
Step 7	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol udp`	Configures the match criteria for a class map on the basis of a specified protocol.
Step 8	**exit** **Example:** `Router(config-cmap)# exit`	Exits class map configuration mode and enters global configuration mode.
Step 9	**class-map match-any** *class-map-name* **Example:** `Router(config)# class-map match-any parent`	Creates a Layer 3 or Layer 4 class map and enters class map configuration mode.

	Command or Action	Purpose
Step 10	**match class-map** *class-map-name* **Example:** `Router(config-cmap)# match class-map child1`	Configures a traffic class as a classification policy.
Step 11	**match class-map** *class-map-name* **Example:** `Router(config-cmap)# match class-map child2`	Configures a traffic class as a classification policy.
Step 12	**end** **Example:** `Router(config-cmap)# end`	Exits class map configuration mode and enters privileged EXEC mode.

Configuring a Policy Map for a Nested Class 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**
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.
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect pmap`	Creates a Layer 3 or Layer 4 inspect type policy map and enters policy map configuration mode.
Step 4	**class-type inspect** *class-map-name* **Example:** `Router(config-pmap)# class-type inspect parent`	Specifies the traffic (class) on which an action is to be performed and enters policy-map class configuration mode.

	Command or Action	Purpose
Step 5	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS XE stateful packet inspection.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Exits policy-map class configuration mode and enters privileged EXEC mode.

Attaching a Policy Map to a Zone Pair

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *zone-name* **destination** [*zone-name*]]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **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-name* **Example:** `Router(config)# zone security source-zone`	Creates a security zone to which interfaces can be assigned 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-name* **Example:** `Router(config)# zone security destination-zone`	Creates a security zone to which interfaces can be assigned 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-pair security** *zone-pair-name* [**source** *zone-name* **destination** [*zone-name*]] **Example:** `Router(config)# zone-pair security secure-zone source source-zone destination destination-zone`	Creates a zone pair and enters security zone pair configuration mode. • To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect pmap`	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 9	**exit** **Example:** `Router(config-sec-zone-pair)# exit`	Exits security zone pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security source-zone`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for Nested Class Map Support for Zone-Based Policy Firewall

Example: Configuring a Two-Layer Nested Class Map

```
Router# configure terminal
Router(config)# class-map match-any child1
Router(config-cmap)# match protocol tcp
Router(config-cmap)# exit
Router(config)# class-map match-any child2
Router(config-cmap)# match protocol udp
Router(config-cmap)# exit
Router(config)# class-map match-any parent
Router(config-cmap)# match class-map child1
Router(config-cmap)# match class-map child2
Router(config-cmap)# end
```

Example: Configuring a Policy Map for a Nested Class Map

```
Router# configure terminal
Router(config)# policy-map type inspect pmap
Router(config-pmap)# class-type inspect parent
Router(config-pmap-c)# inspect
Router(config-pmap-c)# end
```

Example: Attaching a Policy Map to a Zone Pair

```
Router# configure terminal
Router(config)# zone security source-zone
Router(config-sec-zone)# exit
Router(config)# zone security destination-zone
Router(config-sec-zone)# exit
Router(config)# zone-pair security secure-zone source source-zone destination destination-zone
Router(config-sec-zone-pair)# service-policy type inspect pmap
Router(config-sec-zone-pair)# exit
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# zone-member security source-zone
Router(config-if)# end
```

Additional References for Nested Class Map Support for Zone-Based Policy Firewall

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
Zone-based policy firewall	*Zone-Based Policy Firewall*

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

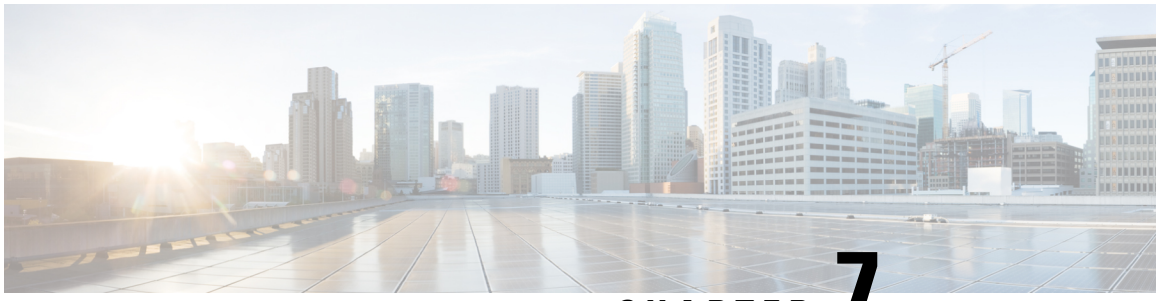
Feature Information for Nested Class Map Support for Zone-Based Policy Firewall

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 8: Feature Information for Nested Class Map Support for Zone-Based Policy Firewall

Feature Name	Releases	Feature Information
Nested Class Map Support for Zone-Based Policy Firewall	Cisco IOS XE Release 3.5S	The Nested Class Map Support for Zone-Based Policy Firewall feature provides the Cisco IOS XE firewall the functionality to configure multiple traffic classes (which are also called nested class maps or hierarchical class maps) as a single traffic class. When packets meet more than one match criterion, you can configure multiple class maps that can be associated with a single traffic policy.



CHAPTER 7

Zone Mismatch Handling

The Zone Mismatch Handling feature allows you to validate the zone pair that is associated with an existing session and allows traffic that matches the zone pair into the network. Allowing traffic into the network without validating the zone pair associated with a session can lead to security vulnerabilities.

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

Finding Feature Information

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

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

Restrictions for Zone Mismatch Handling

You cannot configure the **zone-mismatch drop** command under the **parameter-map type inspect-vrf**, **parameter-map type inspect-zone**, and **parameter-map type inspect global** commands.

Information About Zone Mismatch Handling

Zone Mismatch Handling Overview

The zone-based firewall creates sessions for traffic that flows from a source zone to a destination zone, and also matches the traffic when it returns from the destination zone to the source zone. A zone is a group of interfaces that have similar functions or features. A zone pair allows you to specify a unidirectional firewall policy between two security zones that are part of a zone pair.

For the first packet of the traffic, the firewall checks the zone pair that is associated with the ingress and egress interfaces of the packet, and validates the packet before it creates a session for traffic that can be inspected. And when the return traffic comes, the firewall does a session lookup based on the first packet to find an existing session. If the firewall finds a matching session, it allows the traffic to passthrough, and does not check whether the zone associated with the return traffic matches with the zone pair associated with the existing session. Allowing traffic into the network without validating the zone-pair associated with a session can lead to security vulnerabilities.

The Zone Mismatch Handling feature allows you to validate the zone pair that is associated with an existing session and allows traffic that matches the zone pair into the network. When you configure the **zone-mismatch drop** command, the firewall drops all packets (IPv4 and IPv6) that match an existing session but whose zone pair does not match the zone through which these packets arrive or leave. This feature works along with high availability and In-Service Software Upgrade (ISSU).

When you configure the **zone-mismatch drop** command under the **parameter-map type inspect-global** command, the zone mismatch handling configuration applies to the global firewall configuration. Traffic between all zones are inspected for zone-pair mismatch.

You can also configure the **zone-mismatch drop** command under the **parameter-map type inspect** command. This allows you to apply the Zone-Mismatch Handling feature on a per-policy basis.

When you configure the **zone-mismatch drop** command, the configuration is effective only for new sessions. For existing sessions, traffic is not dropped if the sessions do not belong to the same zone-pair.

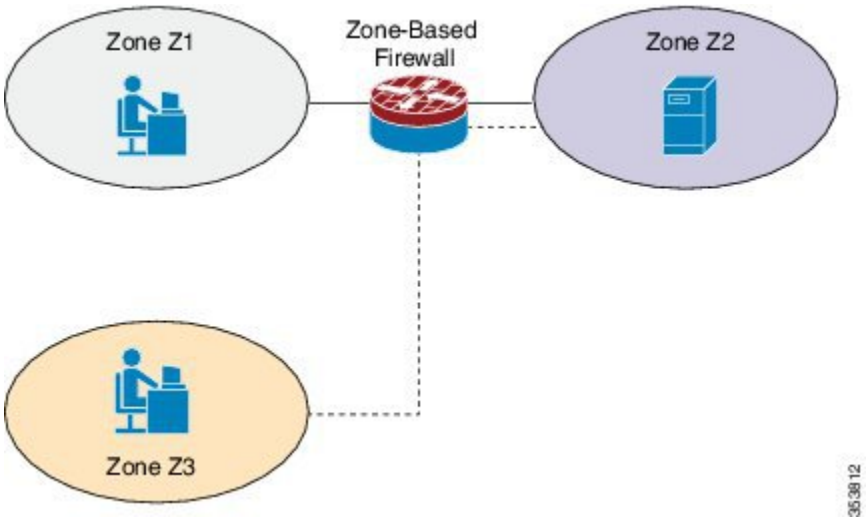
Deployment Scenarios for Zone Mismatch Handling

This section describes some typical scenarios in which the Zone Mismatch Handling feature is deployed:

Traffic Inspection by the Zone-Based Firewall

The following illustration shows traffic inspection by the firewall when the Zone Mismatch Handling feature is enabled.

Figure 11: Traffic Inspection by the Zone-Based Firewall



Zones Z1 and Z2 are part of the same zone pair, which has a parameter map that has the **zone-mismatch drop** command configured on it. Because zone Z3 is not part of the zone pair, the traffic from Z3 is dropped even if the traffic matches the firewall sessions between interface 1 and interface 2.

If you configure the **zone-mismatch drop** command for the parameter-map that is associated with the zone pair to which zone Z3 is attached, that configuration will not be effective for sessions established between Z1 and Z2. However, if you configure the **zone-mismatch drop** command under the **parameter-map type inspect-global** command, the configuration is effective for traffic between all the zones.

Application Layer Gateways Configured with the Zone-Based Firewall

Some application layer gateways (ALGs) also called application-level gateways require multiple control and media channels to operate. The zone-based firewall does not enforce that control and media channels should be in the same zone pair for ALGs. When you configure the **zone-mismatch drop** command for media or data channels, the configuration takes effect after the media or data channels are promoted from imprecise to precise sessions. The zone-based firewall checks these precise sessions like normal sessions. Imprecise sessions are sessions that do not have all 5-tuple information.

How to Configure Zone Mismatch Handling

Configuring Zone Mismatch Handling

You cannot configure the zone-mismatch drop command under the **parameter-map type inspect-vrf**, **parameter-map type inspect-zone**, and **parameter-map type inspect global** commands.

If you configure the **zone-mismatch drop** command under the **parameter-map type inspect-global**command, the zone mismatch handling configuration applies to the global firewall configuration.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. Do one of the following:
   - **parameter-map type inspect** *parameter-map-name*
   - **parameter-map type inspect-global**
4. **zone-mismatch drop**
5. **end**

DETAILED STEPS

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

Step 3	Do one of the following:
	• **parameter-map type inspect** *parameter-map-name*
	• **parameter-map type inspect-global**
	Example:
	`Device(config)# parameter-map type inspect pmap1`
	`or`
	`Device(config)# parameter-map type inspect-global`
		Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the inspect action and enters parameter-map type inspect configuration mode.

Step 4	**zone-mismatch drop** **Example:** `Device(config-profile)# zone-mismatch drop`	Validates the zone pair that is attached to an existing session and allows traffic that matches the zone pair into the network. If the zone pair of an incoming session does not match the zone through which the session arrives or leaves, the firewall drops these packets.
Step 5	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Configuration Examples for Zone Mismatch Handling

Example: Configuring Zone Mismatch Handling

In the following example, the Zone Mismatch Handling feature is enabled for parameter map pmap-fw.

```
! Configuring zones
Device(config)# zone security private
Device(config-sec-zone)# exit
```

```
Device(config)# zone security public
Device(config-sec-zone)# exit
Device(config)# zone security internet
Device(config-sec-zone)# exit

! Attaching zones to interfaces
Device(config)# interface GigabitEthernet 0/1/5
Device(config-if)# ip address 172.16.1.1 255.255.255.0
Device(config-if)# zone-member security private
Device(config-if)# no shutdown
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/1/6
Device(config-if)# ip address 209.165.200.226 255.255.255.0
Device(config-if)# zone-member security public
Device(config-if)# no shutdown
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/1/1
Device(config-if)# ip address 198.51.100.1 255.255.255.0
Device(config-if)# zone-member security internet
Device(config-if)# no shutdown
Device(config-if)# exit

!Configuring the Zone Mismatch Handling feature
Device(config)# parameter-map type inspect pmap-fw
Device(config-profile)# zone-mismatch drop
Device(config-profile)# exit

!Configuring class maps
Device(config)# class-map type inspect match-any internet-traffic-class
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol udp
Device(config-cmap)# match protocol icmp
Device(config-cmap)# exit

! Configuring policy maps and class matching
Device(config)# policy-map type inspect private-internet-policy
Device(config-pmap)# class type inspect internet-traffic-class
Device(config-pmap-c)# inspect pmap-fw
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
Device(config-pmap-c)# exit
Device(config-pmap)# exit

! Configuring zone pairs
Device(config)# zone-pair security private-internet source private destination internet
Device(config-sec-zone-pair)# service-policy type inspect private-internet-policy
Device(config-sec-zone-pair)# end
```

Additional References for Zone Mismatch Handling

Related Documents

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

Related Topic	Document Title
Security Commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support 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

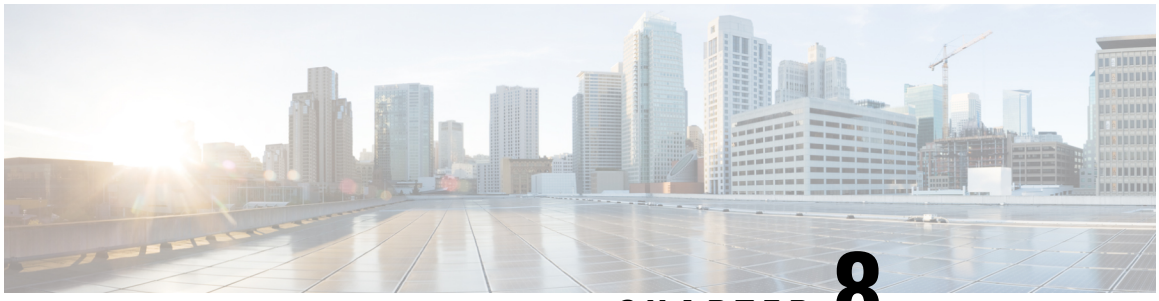
Feature Information for Zone Mismatch Handling

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

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

Table 9: Feature Information for Zone Mismatch Handling

Feature Name	Releases	Feature Information
Zone Mismatch Handling	Cisco IOS XE 3.15S	The Zone Mismatch Handling feature allows you to validate the zone-pair associated with an existing session and allows traffic that matches the zone-pair into the network. This feature is supported on Cisco 4400 Series Integrated Services Routers,Cisco ASR 1000 Series Aggregation Services Routers, and Cisco Cloud Services Router 1000V Series. The following command was introduced: **zone-mismatch handling**.



CHAPTER 8

Configuring Firewall Stateful Interchassis Redundancy

The Firewall Stateful Interchassis Redundancy feature enables you to configure pairs of routers to act as backup for each other. This feature 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 performing traffic forwarding services and maintaining a dynamic routing table.

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 Firewall Stateful Interchassis Redundancy

- The interfaces attached to the firewall must have the same redundant interface identifier (RII).
- The active device and the standby device must have the same Cisco IOS XE Zone-Based Firewall configuration.
- The active device and the standby device must run on an identical version of the Cisco IOS XE software. The active device and the standby device must be connected through a switch.

- Embedded Service Processor (ESP) must match on both active and standby devices.

Restrictions for Firewall Stateful Interchassis Redundancy

- LAN and MESH scenarios are not supported.
- Cisco ASR 1006 and Cisco ASR 1013 platforms with dual Embedded Services Processors (ESPs) or dual Route Processors (RPs) in the chassis are not supported, because coexistence of interbox high availability (HA) and intrabox HA is not supported.

  Cisco ASR 1006 and Cisco ASR 1013 platforms with single ESP and single RP in the chassis supports interchassis redundancy.
- If the dual IOS daemon (IOSd) is configured, the device will not support the firewall Stateful Interchassis Redundancy configuration.

Information About Firewall Stateful Interchassis Redundancy

How Firewall Stateful Inter-Chassis Redundancy Works

You can configure pairs of routers to act as hot standbys for each other. This redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups. The figure below depicts the active-standby device scenario. It shows how the redundancy group is configured for a pair of routers that has one outgoing interface. The *Redundancy Group Configuration--Two Outgoing Interfaces* figure depicts the active-active device scenario shows how two redundancy groups are configured for a pair of routers that have two outgoing interfaces.

Note that in both cases, the redundant routers are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of the routers. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and to synchronize the stateful database for these applications.

Also, in both cases, the pairs of redundant interfaces are configured with the same unique ID number known as the RII.

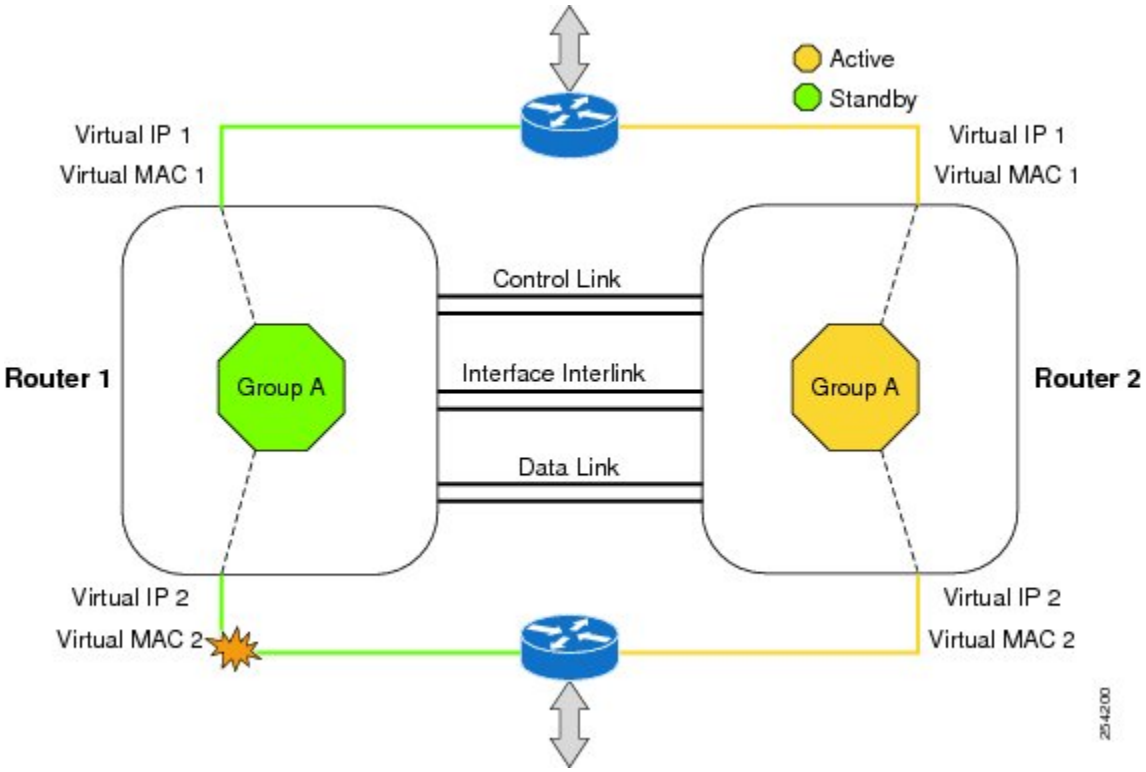
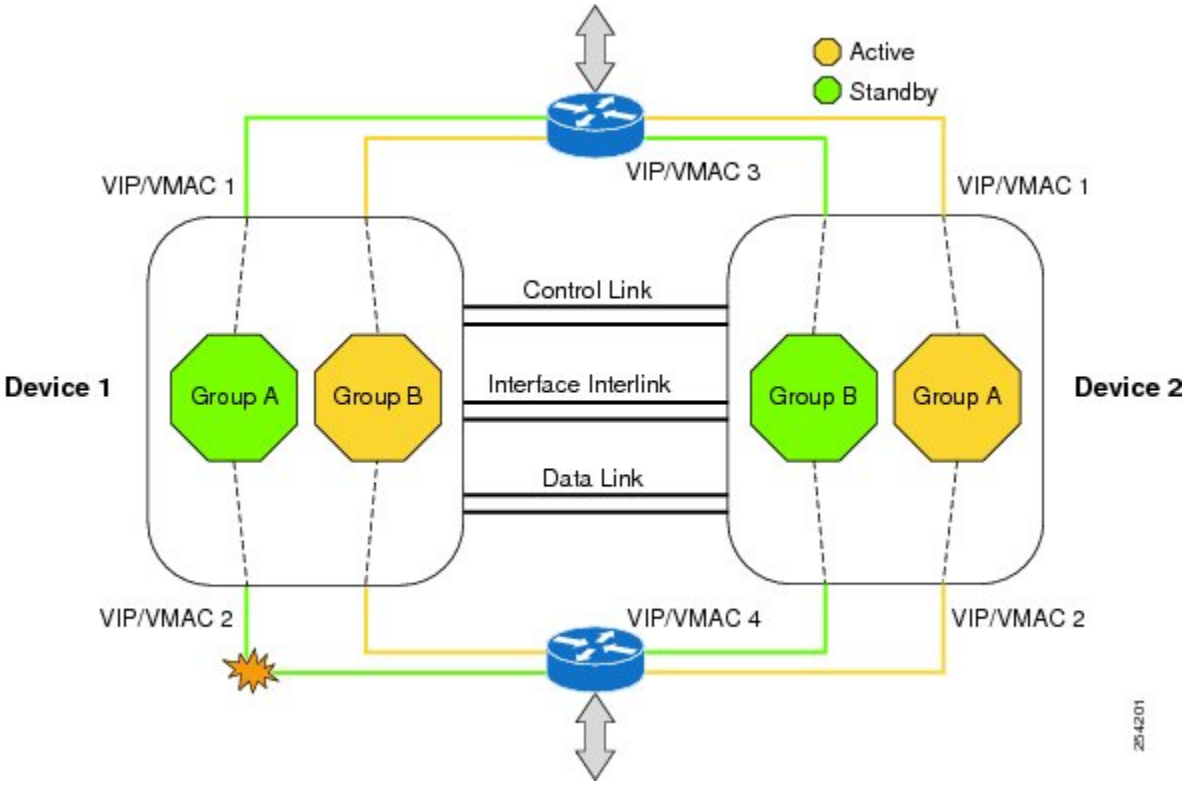


Figure 12: Redundancy Group Configuration--Two Outgoing Interfaces



The status of redundancy group members is determined through the use of hello messages sent over the control link. If either of the routers does not respond to a hello message within a configurable amount of time, it is considered that a failure has occurred, and a switchover is initiated. To detect a failure in milliseconds, the control links run the failover protocol integrated with the Bidirectional Forwarding Detection (BFD) protocol. You can configure the following parameters for the hello messages:

- Active timer
- Standby timer
- Hellotime--The interval at which hello messages are sent
- Holdtime--The amount of time before the active or the standby router is declared to be down

The hellotime defaults to 3 seconds to align with Hot Standby Router Protocol (HSRP), and the holdtime 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 number for each pair of redundant interfaces. This ID number is known as the RII associated with the interface.

A switchover to the standby router can also occur under other circumstances. Another factor that can cause a switchover is a priority setting that is configurable for each router. The router with the highest priority value will be the active router. If a fault occurs on either the active or the standby router, the priority of the router is decremented by a configurable amount known as the weight. If the priority of the active router falls below the priority of the standby router, a switchover occurs and the standby router becomes the active router. 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 L1 state of the interface goes down. This amount overrides the default amount configured for the 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, previous priority, new priority, and a description of the failure event cause.

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

In general, a switchover to the standby router occurs under the following circumstances:

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

In the Firewall Stateful Inter-Chassis Redundancy feature, the redundancy group traffic is routed through the virtual IP address that is associated with the ingress interface of the redundancy group. The traffic sent to the

virtual IP address is received by the router that has the redundancy group in the active state. During a redundancy group failover, the traffic to the virtual IP address is automatically routed to the newly active redundancy group.

The firewall drops the traffic that arrives on the standby redundancy group in case the redundancy group traffic is routed through the physical IP address of a standby router and the traffic reaches the standby redundancy group. However, when the traffic arrives on the active redundancy group, the established TCP or UDP sessions are synchronized to the standby redundancy group.

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.

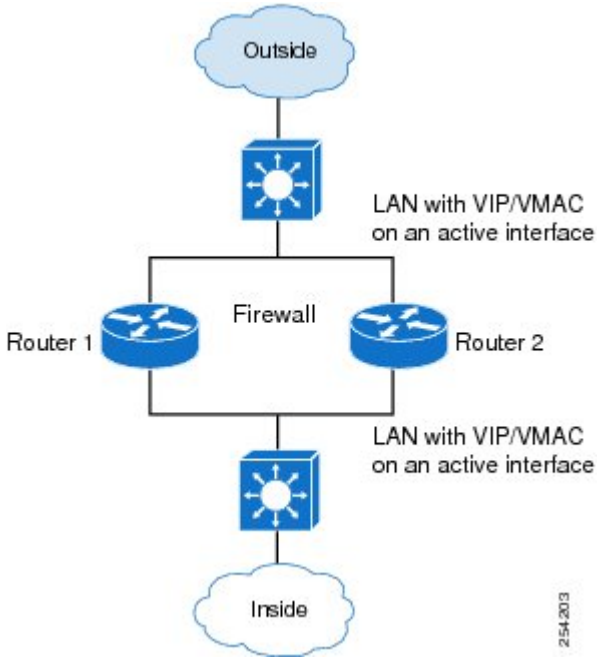
You must configure a physical IP address before configuring an IPv4 VIP.

Supported Topologies

The LAN-LAN topology is supported in the Firewall Stateful Inter-Chassis Redundancy architecture:

LAN-LAN

The figure below shows the LAN-LAN topology. When a dedicated appliance-based firewall solution is used, traffic is often directed to the correct firewall by configuring static routing in the upstream or downstream routers to an appropriate virtual IP address. In addition, the Aggregation Services Routers (ASRs) will participate in dynamic routing with upstream or downstream routers. 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.



For more information about the LAN-LAN configuration, see the section, Example Configuring LAN-LAN.

VRF-Aware Interchassis Redundancy in Zone-Based Firewalls

In Cisco IOS XE Release 3.14S, zone-based firewalls support VRF-aware interchassis redundancy. The VPN routing and forwarding (VRF) name at the active and standby devices must the same. The same VRF configuration must be available on both active and standby devices.

The VRF-Aware Interchassis Redundancy in Zone-Based Firewalls feature uses a VRF mapping mechanism that sends the VRF hash key along with box-to-box high availability session sync messages across active and standby devices.

How to Configure Firewall Stateful Interchassis Redundancy

Configuring a Redundancy Application Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **shutdown**
8. **priority** *value* [**failover threshold** *value*]

9. **preempt**
10. **track** *object-number* {**decrement** *value* | **shutdown**}
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`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	(Optional) Specifies an optional alias for the protocol instance.
Step 7	**shutdown** **Example:** `Device(config-red-app-grp)# shutdown`	(Optional) Shuts down a redundancy group manually.
Step 8	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	(Optional) Specifies the initial priority and failover threshold for a redundancy group.
Step 9	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the group and enables the standby device to preempt the active device regardless of the priority.
Step 10	**track** *object-number* {**decrement** *value* | **shutdown**} **Example:**	Specifies the priority value of a redundancy group that will be decremented if an event occurs.

	Command or Action	Purpose
	`Device(config-red-app-grp)# track 200 decrement 200`	
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Redundancy Group Protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **protocol** *id*
6. **name** *group-name*
7. **timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*}
8. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-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	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**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.

	Command or Action	Purpose
Step 6	**name** *group-name* **Example:** `Device(config-red-app-prtcl)# name prot1`	(Optional) Configures the redundancy group (RG) with a name.
Step 7	**timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9`	Specifies the interval between when hello messages are sent and the time period before which a device is declared to be down.
Step 8	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-name*} **Example:** `Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`	Specifies the authentication information.
Step 9	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring a Virtual IP Address and a Redundant Interface Identifier

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **redundancy rii** *id*
5. **redundancy group** *id* **ip** *virtual-ip* **exclusive** [**decrement** *value*]
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	**interface** *type number* **Example:**	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface GigabitEthernet 0/1/1`	
Step 4	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 600`	Configures the redundancy interface identifier (RII) for a redundancy group. • The range is from 1 to 65535.
Step 5	**redundancy group** *id* **ip** *virtual-ip* **exclusive** [**decrement** *value*] **Example:** `Device(config-if)# redundancy group 1 ip 10.10.1.1 exclusive decrement 20`	Associates an interface with a redundancy group and enables a virtual IP address.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a Control Interface and a Data 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. **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.

	Command or Action	Purpose
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/0`	Specifies the data interface that is used by the redundancy group.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/2 protocol 1`	Specifies the control interface that is used by the redundancy group. • This 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 that a redundancy group will take to delay role negotiations that start after a fault occurs or the system is reloaded.
Step 9	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Managing and Monitoring Firewall Stateful Inter-Chassis Redundancy

Use the following commands to manage and monitor the Firewall Stateful Inter-Chassis Redundancy feature.

SUMMARY STEPS

1. **enable**
2. **debug redundancy application group config** {**all** | **error** | **event** | **func**}
3. **debug redundancy application group faults** {**all** | **error** | **event** | **fault** | **func**}
4. **debug redundancy application group media** {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**}
5. **debug redundancy application group protocol** {**all** | **detail** | **error** | **event** | **media** | **peer**}
6. **debug redundancy application group rii** {**error** | **event**}
7. **debug redundancy application group transport** {**db** | **error** | **event** | **packet** | **timer** | **trace**}
8. **debug redundancy application group vp** {**error** | **event**}
9. **show redundancy application group** [*group-id* | **all**]
10. **show redundancy application transport** {**client** | **group** [*group-id*]}
11. **show redundancy application control-interface group** [*group-id*]

12. **show redundancy application faults group** [*group-id*]
13. **show redundancy application protocol** {*protocol-id* | **group** [*group-id*]
14. **show redundancy application if-mgr group** [*group-id*]
15. **show redundancy application data-interface group** [*group-id*]
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug redundancy application group config** {**all** | **error** | **event** | **func**} **Example:** `Device# debug redundancy application group config all`	Displays the redundancy group application configuration.
Step 3	**debug redundancy application group faults** {**all** | **error** | **event** | **fault** | **func**} **Example:** `Device# debug redundancy application group faults error`	Displays the redundancy group application fault.
Step 4	**debug redundancy application group media** {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**} **Example:** `Device# debug redundancy application group media timer`	Displays the redundancy group application group media information.
Step 5	**debug redundancy application group protocol** {**all** | **detail** | **error** | **event** | **media** | **peer**} **Example:** `Device# debug redundancy application group protocol peer`	Displays the redundancy group application group protocol information.
Step 6	**debug redundancy application group rii** {**error** | **event**} **Example:** `Device# debug redundancy application group rii event`	Displays the redundancy group application group RII information.

	Command or Action	Purpose
Step 7	**debug redundancy application group transport** {**db** | **error** | **event** | **packet** | **timer** | **trace**} **Example:** `Device# debug redundancy application group transport trace`	Displays the redundancy group application group transport information.
Step 8	**debug redundancy application group vp** {**error** | **event**} **Example:** `Device# debug redundancy application group vp event`	Displays the redundancy group application group VP information.
Step 9	**show redundancy application group** [*group-id* | **all**] **Example:** `Device# show redundancy application group all`	Displays the redundancy group information.
Step 10	**show redundancy application transport** {**client** | **group** [*group-id*]} **Example:** `Device# show redundancy application transport group 1`	Displays transport specific information for a redundancy group.
Step 11	**show redundancy application control-interface group** [*group-id*] **Example:** `Device# show redundancy application control-interface group 2`	Displays control interface information for a redundancy group.
Step 12	**show redundancy application faults group** [*group-id*] **Example:** `Device# show redundancy application faults group 2`	Displays fault-specific information for a redundancy group.
Step 13	**show redundancy application protocol** {*protocol-id* | **group** [*group-id*] **Example:** `Device# show redundancy application protocol 3`	Displays protocol specific information for a redundancy group.
Step 14	**show redundancy application if-mgr group** [*group-id*] **Example:** `Device# show redundancy application if-mgr group 2`	Displays interface manager information for a redundancy group.

	Command or Action	Purpose
Step 15	**show redundancy application data-interface group** [*group-id*] **Example:** `Device# show redundancy application data-interface group 1`	Displays data interface specific information.
Step 16	**end** **Example:** `Device# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuration Examples for Firewall Stateful Interchassis Redundancy

Example: Configuring a Redundancy Application Group

The following example shows how to configure a redundancy group named group1 with priority and preempt attributes:

```
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 200 decrement 200
Device(config-red-app-grp)# end
```

Example: Configuring a Redundancy Group Protocol

The following example shows how to configure a redundancy group with timers set for hello time and hold time messages:

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9
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 a Virtual IP Address and a Redundant Interface Identifier

The following example shows how to configure the redundancy group virtual IP address for Gigabit Ethernet interface 0/1/1:

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/1
Device(conf-if)# redundancy rii 600
Device(config-if)# redundancy group 2 ip 10.2.3.4 exclusive decrement 200
Device(config-if)# end
```

Example: Configuring a Control Interface and a Data Interface

```
Device# configure terminal
Device(config-red)# application redundancy
Device(config-red-app-grp)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/0
Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# end
```

Example: Configuring a LAN-LAN Topology

The following is a sample LAN-LAN configuration that shows how a pair of routers that have two outgoing interfaces are configured for stateful redundancy. In this example, GigabitEthernet 0/1/1 is the ingress interface and GigabitEthernet 0/2/1 is the egress interface. Both interfaces are assigned to zones and a classmap is defined to describe the traffic between zones. Interfaces are also configured for redundancy. The “inspect” action invokes the application-level gateway (ALG) to open a pinhole to allow traffic on other ports. A pinhole is a port that is opened through an ALG to allow a particular application to gain controlled access to a protected network.

The following is the configuration on Device 1, the active device.

```
! Configures redundancy, control and data interfaces
redundancy
 mode none
 application redundancy
  group 2
   preempt
   priority 200 failover threshold 100
   control GigabitEthernet 0/0/4 protocol 2
   data GigabitEthernet 0/0/3
!
  protocol 2
   timers hellotime ms 250 holdtime ms 750
!
! Configures a VRF
ip vrf vrf1
!
! Configures parameter maps to add parameters that control the behavior of actions and match
 criteria.
parameter-map type inspect pmap-udp
 redundancy
  redundancy delay 10
!
parameter-map type inspect pmap-tcp
 redundancy
```

```
  redundancy delay 10
!
! Defines class-maps to describes traffic between zones
class-map type inspect match-any cmap-udp
 match protocol udp
!
class-map type inspect match-any cmap-ftp-tcp
 match protocol ftp
 match protocol tcp
!
! Associates class-maps with policy-maps to define actions to be applied
policy-map type inspect p1
 class type inspect cmap-udp
  inspect pmap-udp
!
 class type inspect cmap-ftp-tcp
  inspect pmap-tcp
!
! Identifies and defines network zones
zone security z-int
!
zone security z-hi
!
! Sets zone pairs for any policy other than deny all and assign policy-maps to zone-pairs
by defining a service-policy
zone-pair security hi2int source z-hi destination z-int
 service-policy type inspect p1
!
! Assigns interfaces to zones
interface GigabitEthernet 0/0/1
 ip vrf forwarding vrf1
 ip address 10.1.1.3 255.255.0.0
 ip virtual-reassembly
 zone-member security z-hi
 negotiation auto
 redundancy rii 20
 redundancy group 2 ip 10.1.1.10 exclusive decrement 50
!
interface GigabitEthernet 0/0/2
 ip vrf forwarding vrf1
 ip address 192.0.2.2 255.255.255.240
 ip virtual-reassembly
 zone-member security z-int
 negotiation auto
 redundancy rii 21
 redundancy group 2 ip 192.0.2.12 exclusive decrement 50
!
interface GigabitEthernet 0/0/4
 ip address 198.51.100.17 255.255.255.240
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.49 255.255.255.240
!
ip route vrf vrf1 192.0.2.0 255.255.255.240 GigabitEthernet0/0/2 10.1.1.4
ip route vrf vrf1 10.1.0.0 255.255.0.0 GigabitEthernet0/0/1 10.1.0.4
!
```

The following is the configuration on Device 2, the standby device:

```
! Configures redundancy, control and data interfaces
redundancy
 mode none
 application redundancy
  group 2
   preempt
```

```
   priority 200 failover threshold 100
   control GigabitEthernet 0/0/4 protocol 2
   data GigabitEthernet 0/0/3
!
  protocol 2
   timers hellotime ms 250 holdtime ms 750
!
! Configures a VRF
ip vrf vrf1
!
! Configures parameter maps to add parameters that control the behavior of actions and match
 criteria.
parameter-map type inspect pmap-udp
 redundancy
  redundancy delay 10
!
parameter-map type inspect pmap-tcp
 redundancy
  redundancy delay 10
!
! Defines class-maps to describes traffic between zones
class-map type inspect match-any cmap-udp
 match protocol udp
!
class-map type inspect match-any cmap-ftp-tcp
 match protocol ftp
 match protocol tcp
!
! Associates class-maps with policy-maps to define actions to be applied
policy-map type inspect p1
 class type inspect cmap-udp
  inspect pmap-udp
!
 class type inspect cmap-ftp-tcp
  inspect pmap-tcp
!
! Identifies and defines network zones
zone security z-int
!
zone security z-hi
!
! Sets zone pairs for any policy other than deny all and assign policy-maps to zone-pairs
by defining a service-policy
zone-pair security hi2int source z-hi destination z-int
 service-policy type inspect p1
!
! Assigns interfaces to zones
interface GigabitEthernet 0/0/1
 ip vrf forwarding vrf1
 ip address 10.1.1.6 255.255.0.0
 ip virtual-reassembly
 zone-member security z-hi
 negotiation auto
 redundancy rii 20
 redundancy group 2 ip 10.1.1.12 exclusive decrement 50
!
interface GigabitEthernet 0/0/2
 ip vrf forwarding vrf1
 ip address 192.0.2.5 255.255.255.240
 ip virtual-reassembly
 zone-member security z-int
 negotiation auto
 redundancy rii 21
 redundancy group 2 ip 192.0.2.10 exclusive decrement 50
```

```
!
interface GigabitEthernet 0/0/4
 ip address 198.51.100.21 255.255.255.240
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.53 255.255.255.240
!
ip route vrf vrf1 192.0.2.0 255.255.255.240 GigabitEthernet0/0/2 10.1.1.4
ip route vrf vrf1 10.1.0.0 255.255.0.0 GigabitEthernet0/0/1 10.1.0.4
!
```

Additional References for Firewall Stateful Interchassis Redundancy

Related Documents

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

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

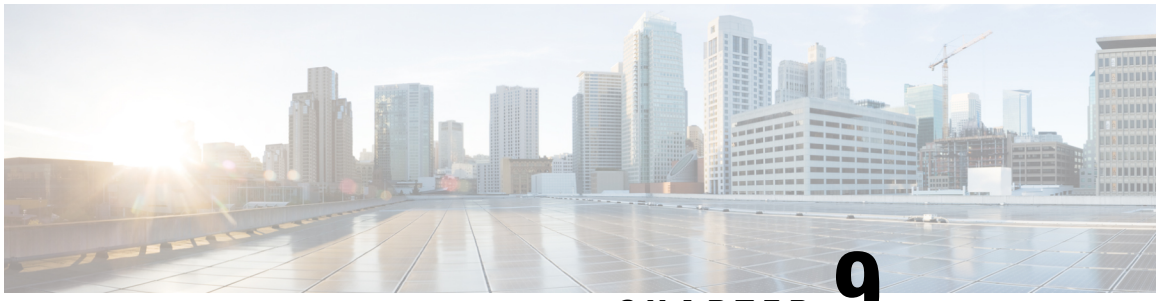
Feature Information for Firewall Stateful Interchassis Redundancy

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 10: Feature Information for Firewall Stateful Interchassis Redundancy

Feature Name	Releases	Feature Information
Firewall Stateful Interchassis Redundancy	Cisco IOS XE Release 3.1(S)	The Firewall Stateful Interchassis Redundancy feature enables you to configure pairs of devices to act a backups for each other. The following commands were introduced or modified: **application redundancy**, **authentication**, **control**, **data**, **debug redundancy application group config**, **debug redundancy application group faults**, **debug redundancy application group media**, **debug redundancy application group protocol**, **debug redundancy application group rii**, **debug redundancy application group transport**, **debug redundancy application group vp**, **group**, **name**, **preempt**, **priority**, **protocol**, **redundancy rii**, **redundancy group**, **track**, **timers delay**, **timers hellotime**, **show redundancy application group**, **show redundancy application transport**, **show redundancy application control-interface**, **show redundancy application faults**, **show redundancy application protocol**, **show redundancy application if-mgr**, **show redundancy application data-interface**.
VRF-Aware Stateful Interchassis Redundancy in Zone-Based Firewalls	Cisco IOS XE Release 3.14S	In Cisco IOS XE Release 3.14S, zone-based firewalls support VRF-aware interchassis redundancy. The VPN routing and forwarding (VRF) name at the active and standby devices must the same. The same VRF configuration must be available on both active and standby devices.



CHAPTER 9

Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

The Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls feature supports high availability (HA) based on redundancy groups (RGs) on IPv6 firewalls. This feature enables you to configure pairs of devices to act as backup for each other. This feature can be configured to determine the active device based on a number of failover conditions. This feature supports the FTP66 application-layer gateway (ALG) for IPv6 packet inspection.

This module provides information about Box-to-Box (B2B) HA support and describes how to configure this 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.

Prerequisites for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

- Interfaces attached to a firewall must have the same redundant interface identifier (RII).
- Active and standby devices must have the same zone-based policy firewall configuration.
- Active and standby devices must run on identical versions of Cisco software. The active and standby devices must be connected through a switch.
- The box-to-box (B2B) configuration on both active and standby devices should be the same because there is no autosynchronization of the configuration between these devices.
- For asymmetric routing traffic to pass, you must configure the pass action for the class-default class. Class-default class is a system-defined class map that represents all packets that do not match any of the user-defined classes in a policy.
- 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 Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

- Only IPv4 is supported at box-to-box (B2B) interlink interfaces.
- Multiprotocol Label Switching (MPLS) and virtual routing and forwarding (VRF) are not supported.
- Cisco ASR 1006 and 1013 Aggregation Services Routers with dual Embedded Services Processors (ESPs) or dual Route Processors (RPs) in the chassis are not supported, because coexistence of interbox high availability (HA) and intrabox HA is not supported.

  Cisco ASR 1006 and Cisco ASR 1013 Aggregation Services Routers with single ESP and single RP in the chassis support interchassis redundancy.
- If the dual IOS daemon (IOSd) is configured, the device will not support the firewall stateful interchassis redundancy configuration.
- Stateless Network Address Translation 64 (NAT64) with IPv6 firewalls is not supported.

Information About Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

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.

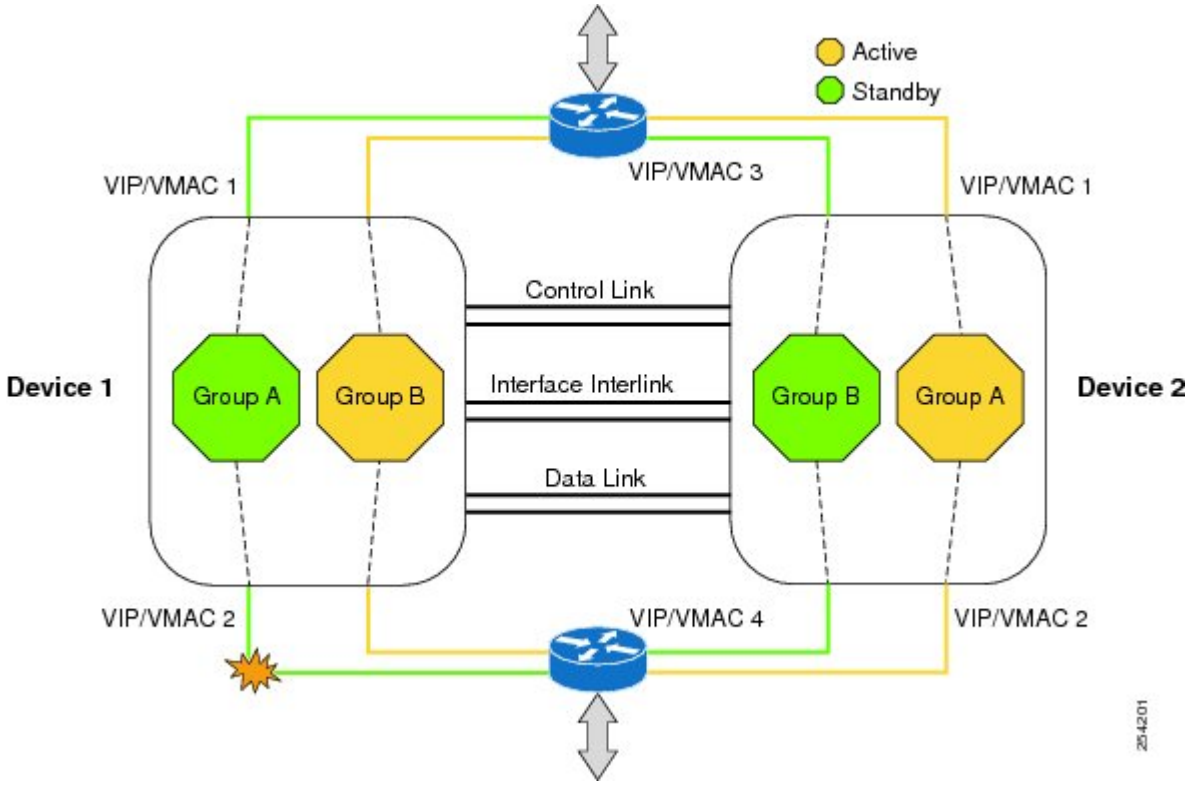
When active/standby high availability failover is configured, only one of the devices involved in the failover handles the traffic at one time, while the other device is in a standby mode, periodically synchronizing session information from the active device.

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.

Box-to-Box High Availability Operation

You can configure pairs of devices to act as hot standbys for each other. Redundancy is configured per interface. Pairs of redundant interfaces are known as redundancy groups (RGs). Figure 1 depicts an active/active failover scenario. It shows how two redundancy groups are configured for a pair of devices that have two outgoing interfaces.

Figure 13: Redundancy Group Configuration—Two Outgoing Interfaces



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). The routing table is not synced from active to standby.

Asymmetric routing is supported as part of the firewall HA. 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.

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 configured time period, 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 time period before which the active or standby device is declared to be down.

The hello time defaults to three 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 the RII that is associated with the interface.

Reasons for Switchover

Another factor that can cause a switchover is the 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. You can override this default behavior 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 the 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 the 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 below that of the standby device.
- The run-time priority of the active device goes below the configured threshold level.
- 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.

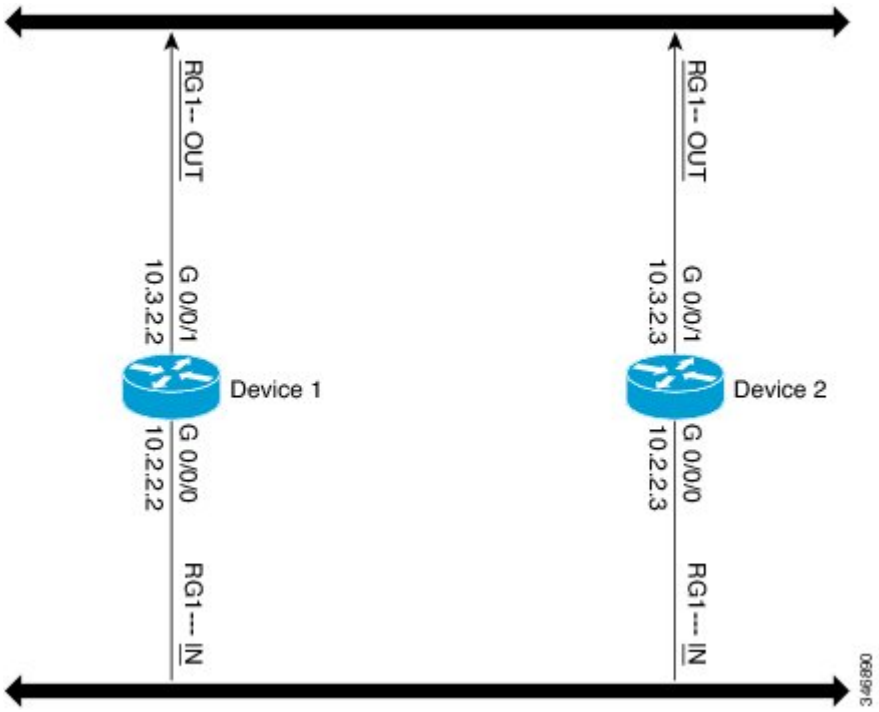
In an active/standby scenario, the main difference between two devices in a failover pair depends on which device is active and which device is a standby, namely which IP addresses to use and which device actively passes the traffic. The active device always becomes the active device if both devices 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.

NAT Box-to-Box High-Availability 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. The figure below shows the NAT box-to-box LAN-LAN topology. Network Address Translation (NAT) is in the active-standby mode and the peers are in one redundancy group (RG). All traffic or a subset of this traffic undergoes NAT translation.

Note Failover is caused by only those failures that the RG infrastructure listens to.

Figure 14: NAT Box-to-Box High-Availability LAN-LAN Topology



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.

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.

FTP66 ALG Support Overview

Firewalls support the inspection of IPv6 packets and stateful Network Address Translation 64 (NAT64). For FTP to work over IPv6 packet inspection, the application-layer gateway (ALG) (also called the application-level gateway [ALG]), FTP66, is required. The FTP66 ALG is also called all-in-one FTP ALG and one FTP ALG.

The FTP66 ALG supports the following:

- Firewall IPv4 packet inspection
- Firewall IPv6 packet inspection
- NAT configuration
- NAT64 configuration (along with FTP64 support)
- NAT and firewall configuration
- NAT64 and firewall configuration

The FTP66 ALG has the following security vulnerabilities:

- Packet segmentation attack—The FTP ALG state machine can detect segmented packets, and the state machine processing is stopped until a complete packet is received.
- Bounce attack—The FTP ALG does not create doors (for NAT) or pinholes (for firewalls) with a data port number less than 1024. The prevention of a bounce attack is activated only when the firewall is enabled.

How to Configure Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Configuring a Redundancy Group Protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **protocol** *id*
6. **name** *group-name*
7. **timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*}
8. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-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	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:**	Enters redundancy application configuration mode.

	Command or Action	Purpose
	`Device(config-red)# application redundancy`	
Step 5	**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 6	**name** *group-name* **Example:** `Device(config-red-app-prtcl)# name prot1`	(Optional) Configures the redundancy group (RG) with a name.
Step 7	**timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9`	Specifies the interval between when hello messages are sent and the time period before which a device is declared to be down.
Step 8	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-name*} **Example:** `Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`	Specifies the authentication information.
Step 9	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring a Redundancy Application Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **shutdown**
8. **priority** *value* [**failover threshold** *value*]
9. **preempt**
10. **track** *object-number* {**decrement** *value* | **shutdown**}
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`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	(Optional) Specifies an optional alias for the protocol instance.
Step 7	**shutdown** **Example:** `Device(config-red-app-grp)# shutdown`	(Optional) Shuts down a redundancy group manually.
Step 8	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	(Optional) Specifies the initial priority and failover threshold for a redundancy group.
Step 9	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the group and enables the standby device to preempt the active device regardless of the priority.
Step 10	**track** *object-number* {**decrement** *value* | **shutdown**} **Example:** `Device(config-red-app-grp)# track 200 decrement 200`	Specifies the priority value of a redundancy group that will be decremented if an event occurs.
Step 11	**end** **Example:**	Exits redundancy application group configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-red-app-grp)# end`	

Configuring a Control Interface and a Data 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. **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`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/0`	Specifies the data interface that is used by the redundancy group.

	Command or Action	Purpose
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/2 protocol 1`	Specifies the control interface that is used by the redundancy group. • This 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 that a redundancy group will take to delay role negotiations that start after a fault occurs or the system is reloaded.
Step 9	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a LAN Traffic 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. **ipv6 address** {*ipv6-prefix/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **zone-member security** *zone-name*
9. **redundancy rii** *RII-identifier*
10. **redundancy group** *id* {**ip** *virtual-ip* | **ipv6** {*link-local-address* | *ipv6-address/prefix-length*} | **autoconfig**} [**exclusive**] [**decrement** *value*]
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.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/0/2`	Configures an interface and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Device(config-if)# description lan interface`	(Optional) Adds a description to an interface configuration.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if)# encapsulation dot1q 18`	Sets the encapsulation method used by the interface.
Step 6	**ip vrf forwarding** *name* **Example:** `Device(config-if)# 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	**ipv6 address** {*ipv6-prefix/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 8	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone member security z1`	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-if)# redundancy rii 100`	Configures an RII for redundancy group-protected traffic interfaces.
Step 10	**redundancy group** *id* {**ip** *virtual-ip* | **ipv6** {*link-local-address* | *ipv6-address/prefix-length*} | **autoconfig**} [**exclusive**] [**decrement** *value*] **Example:**	Enables the redundancy group (RG) traffic interface configuration.

	Command or Action	Purpose
	`Device(config-if)# redundancy group 1 ipv6 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64 exclusive decrement 50`	
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring a WAN Traffic Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **description** *string*
5. **ipv6 address** {*ipv6-prefix/prefix-length* | *prefix-name sub-bits/prefix-length*}
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 2/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Device(config-if)# description wan interface`	(Optional) Adds a description to an interface configuration.

	Command or Action	Purpose
Step 5	**ipv6 address** {*ipv6-prefix/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8:2222::/48`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z2`	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-if)# 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-if)# redundancy rii 360`	Configures an RII for redundancy group-protected traffic interfaces.
Step 9	**redundancy asymmetric-routing enable** **Example:** `Device(config-if)# redundancy asymmetric-routing enable`	Associates a redundancy group with an interface that is used for asymmetric routing.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring an IPv6 Firewall

The steps to configure an IPv4 firewall and an IPv6 firewall are the same. To configure an IPv6 firewall, you must configure the class map in such a way that only an IPv6 address family is matched.

The **match protocol** command applies to both IPv4 and IPv6 traffic and can be included in either an IPv4 policy or an IPv6 policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf-definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **sessions maximum** *sessions*
9. **exit**
10. **ipv6 unicast-routing**
11. **ip port-map** *appl-name* **port** *port-num* **list** *list-name*
12. **ipv6 access-list** *access-list-name*
13. **permit ipv6 any any**
14. **exit**
15. **class-map type inspect match-all** *class-map-name*
16. **match access-group name** *access-group-name*
17. **match protocol** *protocol-name*
18. **exit**
19. **policy-map type inspect** *policy-map-name*
20. **class type inspect** *class-map-name*
21. **inspect** [*parameter-map-name*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf-definition** *vrf-name* **Example:** `Device(config)# vrf-definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:**	Exits VRF address family configuration mode and enters VRF configuration mode.

	Command or Action	Purpose
	`Device(config-vrf-af)# exit-address-family`	
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect ipv6-param-map`	Enables a global inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
Step 8	**sessions maximum** *sessions* **Example:** `Device(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 11	**ip port-map** *appl-name* **port** *port-num* **list** *list-name* **Example:** `Device(config)# ip port-map ftp port 8090 list ipv6-acl`	Establishes a port to application mapping (PAM) by using the IPv6 access control list (ACL).
Step 12	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 13	**permit ipv6 any any** **Example:** `Device(config-ipv6-acl)# permit ipv6 any any`	Sets permit conditions for an IPv6 access list.
Step 14	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 15	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ipv6-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.

	Command or Action	Purpose
Step 16	**match access-group name** *access-group-name* **Example:** `Device(config-cmap)# match access-group name ipv6-acl`	Configures the match criteria for a class map on the basis of the specified ACL.
Step 17	**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 18	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 19	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ipv6-policy`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 20	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ipv6-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 21	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect ipv6-param-map`	Enables stateful packet inspection.
Step 22	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring Zones and Applying Zones to Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*

11. **ipv6 address** *ipv6-address/prefix-length*
12. **encapsulation dot1q** *vlan-id*
13. **zone-member security** *zone-name*
14. **end**
15. **show policy-map type inspect zone-pair sessions**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters 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 and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** *zone-name* **Example:** `Device(config)# zone security z2`	Creates a security zone and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*] **Example:** `Device(config)# zone-pair security in-2-out source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy`	Attaches a policy map to a top-level policy map.
Step 9	**exit** **Example:**	Exits security zone-pair configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-sec-zone-pair)# exit`	
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0.1`	Configures a subinterface and enters subinterface configuration mode.
Step 11	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-subif)# ipv6 address 2001:DB8:2222:7272::72/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface or a subinterface.
Step 12	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 2`	Sets the encapsulation method used by the interface.
Step 13	**zone-member security** *zone-name* **Example:** `Device(config-subif)# zone member security z1`	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. • 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 the 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 14	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.
Step 15	**show policy-map type inspect zone-pair sessions** **Example:** `Device# show policy-map type inspect zone-pair sessions`	Displays the stateful packet inspection sessions created because a policy map is applied on a specified zone pair. • The output of this command displays both IPv4 and IPv6 firewall sessions.

Example

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv4 address and vice versa:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out
```

```
    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:32847=>(209.165.201.2:21) ftp SIS_OPEN
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [37:84]

        Half-open Sessions
         Session 110D930C [2001:DB8:1::104]:32848=>(209.165.201.2:21) ftp SIS_OPENING
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [0:0]
```

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv6 address:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:63=>[2001:DB8:2::102]:63 udp SIS_OPEN
          Created 00:00:02, Last heard 00:00:01
          Bytes sent (initiator:responder) [162:0]
```

Configuration Examples for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Example: Configuring a Redundancy Group Protocol

The following example shows how to configure a redundancy group with timers set for hello time and hold time messages:

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9
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 a Redundancy Application Group

The following example shows how to configure a redundancy group named group1 with priority and preempt attributes:

```
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 200 decrement 200
Device(config-red-app-grp)# end
```

Example: Configuring a Control Interface and a Data Interface

```
Device# configure terminal
Device(config-red)# application redundancy
Device(config-red-app-grp)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/0
Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# end
```

Example: Configuring a LAN Traffic Interface

```
Device# configure terminal
Device(config-if)# interface gigabitethernet 2/0/2
Device(config-if)# description lan interface
Device(config-if)# encapsulation dot1q 18
Device(config-if)# ip vrf forwarding trust
Device(config-if)# ipv6 address 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64
Device(config-if)# zone member security z1
Device(config-if)# redundancy rii 100
Device(config-if)# redundancy group 1 ipv6 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE exclusive
decrement 50
Device(config-if)# end
```

Example: Configuring a WAN Traffic Interface

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

```
Device# configure terminal
Device(config-if)# interface gigabitethernet 2/1/0
Device(config-if)# description wan interface
Device(config-if)# ipv6 address 2001:DB8:2222::/48
Device(config-if)# zone-member security z2
Device(config-if)# ip tcp adjust-mss 1360
Device(config-if)# redundancy rii 360
Device(config-if)# redundancy asymmetric-routing enable
Device(config-if)# end
```

Example: Configuring an IPv6 Firewall

```
Device# configure terminal
Device(config)# vrf-definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# parameter-map type inspect ipv6-param-map
Device(config-profile)# sessions maximum 10000
Device(config-profile)# exit
Device(config)# ipv6 unicast-routing
Device(config)# ip port-map ftp port 8090 list ipv6-acl
Device(config)# ipv6 access-list ipv6-acl
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# class-map type inspect match-all ipv6-class
Device(config-cmap)# match access-group name ipv6-acl
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect ipv6-policy
Device(config-pmap)# class type inspect ipv6-class
Device(config-pmap-c)# inspect ipv6-param-map
Device(config-pmap-c)# end
```

Example: Configuring Zones and Applying Zones to Interfaces

```
Device# configure terminal
Device(config)# zone security z1
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security in-to-out source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0.1
Device(config-if)# ipv6 address 2001:DB8:2222:7272::72/64
Device(config-if)# encapsulation dot1q 2
Device(config-if)# zone member security z1
Device(config-if)# end
```

Additional References for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Related Documents

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

Related Topic	Document Title
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

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

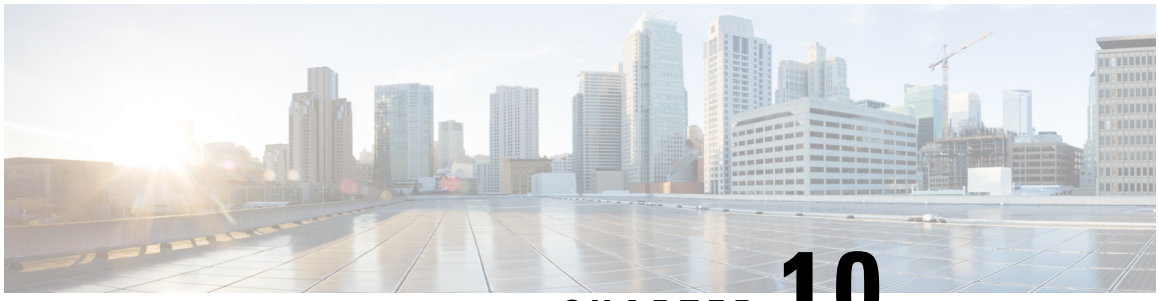
Feature Information for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

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 11: Feature Information for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Feature Name	Releases	Feature Information
Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls	Cisco IOS XE Release 3.8S	The Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls feature supports high availability (HA) based on redundancy groups (RGs) on IPv6 firewalls. This feature enables you to configure pairs of devices to act as backup for each other. This feature can be configured to determine the active device based on a number of failover conditions. No commands were introduced or modified.
Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls	Cisco IOS XE Release 3.8S	In Cisco IOS XE Release 3.10S, support was added for the Cisco ISR 4400 Series Routers.



CHAPTER 10

Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT feature supports the forwarding of packets from a standby redundancy group to the 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.

This module provides an overview of asymmetric routing 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 Firewall and NAT

The following restrictions apply to the Interchassis Asymmetric Routing Support feature:

- LANs that use virtual IP addresses and virtual MAC (VMAC) addresses do not support asymmetric routing.
- In Service Software Upgrade (ISSU) is not supported.

The following features are not supported by the VRF-Aware Asymmetric Routing Support feature:

- Cisco Trustsec
- Edge switching services
- Header compression
- IPsec
- Policy Based Routing (PBR)
- Port bundle
- Lawful intercept
- Layer 2 Tunneling Protocol (L2TP)
- Locator/ID Separation Protocol (LISP) inner packet inspection
- Secure Shell (SSL) VPN
- Session Border Controller (SBC)
- If you enable NAT on the primary and backup WAN link, switchover between the primary and backup interface is not supported. NAT backup interface overload is not supported on the following platforms:
  - ASR1000 Series Aggregation Services Routers
  - ISR4000 Series Integrated Services Routers
  - ISR1000 Series Integrated Services Routers
  - CSR1000 Series Cloud Services Routers

Information About Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

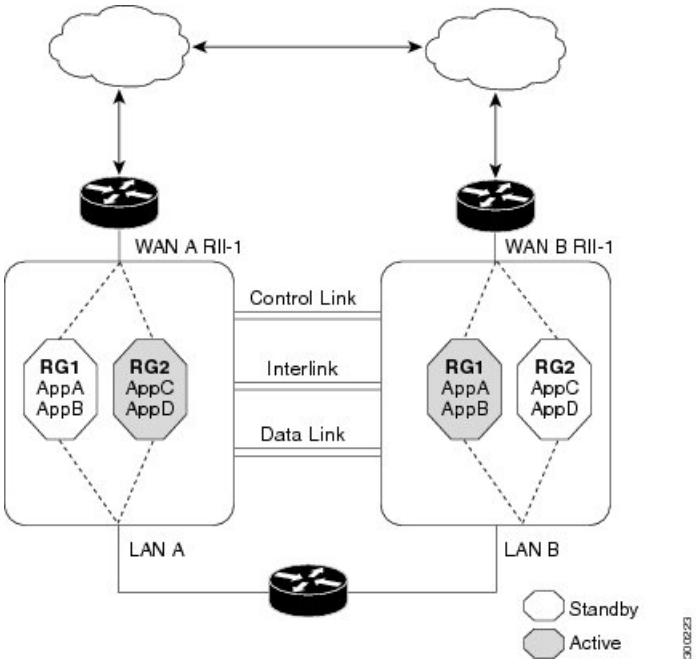
Asymmetric Routing Overview

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 15: 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.

Asymmetric Routing in NAT

By default, when asymmetric routing is configured, Network Address Translation (NAT) processes non-ALG packets on the standby RG, instead of forwarding them to the active. The NAT-only configuration (that is when the firewall is not configured) can use both the active and standby RGs for processing packets. If you have a NAT-only configuration and you have configured asymmetric routing, the default asymmetric routing rule is that NAT will selectively process packets on the standby RG. You can configure the **asymmetric-routing always-divert enable** command to divert packets received on the standby RG to the active RG. Alternatively, if you have configured the firewall along with NAT, the default asymmetric routing rule is to always divert the packets to the active RG.

When NAT receives a packet on the standby RG and if you have not configured the diverting of packets, NAT does a lookup to see if a session exists for that packet. If a session exists and there is no ALG associated for that session, NAT processes the packet on the standby RG. The processing of packets on the standby RG when a session exists significantly increases the bandwidth of the NAT traffic.

ALGs are used by NAT to identify and translate payload and to create child flows. ALGs require a two-way traffic to function correctly. NAT must divert all traffic to the active RG for any packet flow that is associated with an ALG. This is accomplished by checking if ALG data that is associated with the session is found on the standby RG. If ALG data exits, the packet is diverted for asymmetric routing.

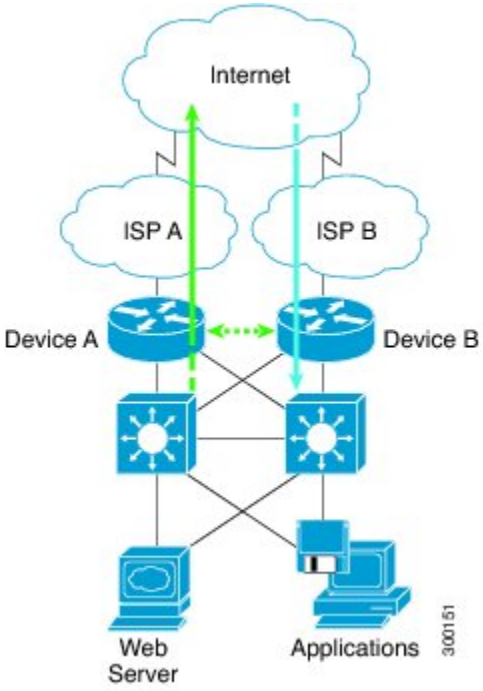
VRF-Aware Software Infrastructure (VASI) support was added in Cisco IOS XE Release 3.16S. Multiprotocol Label Switching (MPLS) asymmetric routing is also supported.

In Cisco IOS XE Release 3.16S, NAT supports asymmetric routing with ALGs, Carrier Grade NAT (CGN), and virtual routing and forwarding (VRF) instances. No configuration changes are required to enable asymmetric routing with ALGs, CGN, or VRF. For more information, see the section, "Example: Configuring Asymmetric Routing with VRF".

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



VRF-Aware Asymmetric Routing in Zone-Based Firewalls

In Cisco IOS XE Release 3.14S, zone-based firewalls support the VRF-Aware Interchassis Asymmetric Routing feature. The feature supports Multiprotocol Label Switching (MPLS).

During asymmetric routing diversion, the VPN routing and forwarding (VRF) name hash value is sent with diverted packets. The VRF name hash value is converted to the local VRF ID and table ID at the active device after the diversion.

When diverted packets reach the active device on which Network Address Translation (NAT) and the zone-based firewall are configured, the firewall retrieves the VRF ID from NAT or NAT64 and saves the VRF ID in the firewall session key.

The following section describes the asymmetric routing packet flow when only the zone-based firewall is configured on a device:

- When MPLS is configured on a device, the VRF ID handling for diverted packets is the same as the handling of non-asymmetric routing diverted packets. An MPLS packet is diverted to the active device, even though the MPLS label is removed at the standby device. The zone-based firewall inspects the packet at the egress interface, and the egress VRF ID is set to zero, if MPLS is detected at this interface. The firewall sets the ingress VRF ID to zero if MPLS is configured at the ingress interface.
- When a Multiprotocol Label Switching (MPLS) packet is diverted to the active device from the standby device, the MPLS label is removed before the asymmetric routing diversion happens.
- When MPLS is not configured on a device, an IP packet is diverted to the active device and the VRF ID is set. The firewall gets the local VRF ID, when it inspects the packet at the egress interface.

VRF mapping between active and standby devices require no configuration changes.

VRF-Aware Asymmetric Routing in NAT

In Cisco IOS XE Release 3.14S, Network Address Translation supports VRF-aware interchassis asymmetric routing. VRF-aware interchassis asymmetric routing uses message digest (MD) 5 hash of the VPN routing and forwarding (VRF) name to identify the VRF and datapath in the active and standby devices to retrieve the local VRF ID from the VRF name hash and viceversa.

For VRF-aware interchassis asymmetric routing, the VRFs on active and standby devices must have the same VRF name. However, the VRF ID need not be identical on both devices because the VRF ID is mapped based on the VRF name on the standby and active devices during asymmetric routing diversion or box-to-box high availability synchronization.

In case of MD5 hash collision for VRF names, the firewall and NAT sessions that belong to the VRF are not synced to the standby device.

VRF mapping between active and standby devices require no configuration changes.

How to Configure Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

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.
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.

	Command or Action	Purpose
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.

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.

	Command or Action	Purpose
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.
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.
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.

	Command or Action	Purpose
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.

Configuring Dynamic Inside Source Translation with Asymmetric Routing

The following configuration is a sample dynamic inside source translation with asymmetric routing. You can configure asymmetric routing with the following types of NAT configurations—dynamic outside source, static inside and outside source, and Port Address Translation (PAT) inside and outside source translations. For more information on different types of NAT configurations, see the "Configuring NAT for IP Address Conservation" chapter.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat outside**
6. **exit**
7. **redundancy**
8. **application redundancy**
9. **group** *id*
10. **asymmetric-routing always-divert enable**
11. **end**
12. **configure terminal**
13. **ip nat pool** *name start-ip end-ip* {*mask* | **prefix-length** *prefix-length*}
14. **exit**
15. **ip nat inside source list** *acl-number* **pool** *name* **redundancy** *redundancy-id* **mapping-id** *map-id*
16. **access-list** *standard-acl-number* **permit** *source-address wildcard-bits*
17. **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`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for an interface.
Step 5	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**redundancy** **Example:** `Device(config)# redundancy`	Configures redundancy and 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 redundancy group and enters redundancy application group configuration mode.
Step 10	**asymmetric-routing always-divert enable** **Example:** `Device(config-red-app-grp)# asymmetric-routing always-divert enable`	Diverts the traffic to the active device.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.
Step 12	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 13	**ip nat pool** *name start-ip end-ip* {*mask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool pool1 prefix-length 24`	Defines a pool of global addresses. • Enters IP NAT pool configuration mode.
Step 14	**exit** **Example:** `Device(config-ipnat-pool)# exit`	Exits IP NAT pool configuration mode and enters global configuration mode.
Step 15	**ip nat inside source list** *acl-number* **pool** *name* **redundancy** *redundancy-id* **mapping-id** *map-id* **Example:** `Device(config)# ip nat inside source list pool pool1 redundancy 1 mapping-id 100`	Enables NAT of the inside source address and associates NAT with a redundancy group by using the mapping ID.
Step 16	**access-list** *standard-acl-number* **permit** *source-address wildcard-bits* **Example:** `Device(config)# access-list 10 permit 10.1.1.1 255.255.255.0`	Defines a standard access list for the inside addresses that are to be translated.
Step 17	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

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
```

Example: Configuring Dynamic Inside Source Translation with Asymmetric Routing

```
Device(config)# interface gigabitethernet 0/1/3
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# end
Device# configure terminal
Device(config)# ip nat pool pool1 prefix-length 24
Device(config-ipnat-pool)# exit
Device(config)# ip nat inside source list pool pool1 redundancy 1 mapping-id 100
Device(config)# access-list 10 permit 10.1.1.1 255.255.255.0
```

Example: Configuring VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

The following is a sample WAN-to-WAN symmetric routing configuration:

```
vrf definition Mgmt-intf
 address-family ipv4
  exit-address-family
!
address-family ipv6
 exit-address-family
!
!
vrf definition VRFA
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 address-family ipv4
  exit-address-family
!
!
no logging console
no aaa new-model
!
multilink bundle-name authenticated
!
redundancy
 mode sso
 application redundancy
  group 1
   preempt
   priority 120
   control GigabitEthernet 0/0/1 protocol 1
   data GigabitEthernet 0/0/2
!
!
!
!
ip tftp source-interface GigabitEthernet0
ip tftp blocksize 8192
!
track 1 interface GigabitEthernet 0/0/4 line-protocol
!
interface Loopback 0
 ip address 209.165.201.1 255.255.255.224
!
interface GigabitEthernet 0/0/0
 vrf forwarding VRFA
 ip address 192.168.0.1 255.255.255.248
 ip nat inside
 negotiation auto
 bfd interval 50 min_rx 50 multiplier 3
 redundancy rii 2
!
interface GigabitEthernet 0/0/1
 ip address 209.165.202.129 255.255.255.224
 negotiation auto
!
interface GigabitEthernet 0/0/2
 ip address 192.0.2.1 255.255.255.224
 negotiation auto
!
interface GigabitEthernet 0/0/3
 ip address 198.51.100.1 255.255.255.240
 negotiation auto
!
interface GigabitEthernet 0/0/4
```

```
 ip address 203.0.113.1 255.255.255.240
 negotiation auto
!
interface GigabitEthernet 0
 vrf forwarding Mgmt-intf
 ip address 172.16.0.1 255.255.0.0
 negotiation auto
!
interface vasileft 1
 vrf forwarding VRFA
 ip address 10.4.4.1 255.255.0.0
 ip nat outside
 no keepalive
!
interface vasiright 1
ip address 10.4.4.2 255.255.0.0
no keepalive
!
router mobile
!
router bgp 577
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 neighbor 203.0.113.1 remote-as 223
 neighbor 203.0.113.1 description PEERING to PTNR neighbor 10.4.4.1 remote-as 577
 neighbor 10.4.4.1 description PEEERING to VASI VRFA interface
!
address-family ipv4
 network 203.0.113.1 mask 255.255.255.240
 network 10.4.0.0 mask 255.255.0.0
 network 209.165.200.224 mask 255.255.255.224
 neighbor 203.0.113.1 activate
 neighbor 10.4.4.1 activate
 neighbor 10.4.4.1 next-hop-self
 exit-address-family
!
address-family ipv4 vrf VRFA
 bgp router-id 4.4.4.4
 network 192.168.0.0 mask 255.255.255.248
 network 10.4.0.0 mask 255.255.0.0
 redistribute connected
 redistribute static
 neighbor 192.168.0.2 remote-as 65004
 neighbor 192.168.0.2 fall-over bfd
 neighbor 192.168.0.2 activate
 neighbor 10.4.4.2 remote-as 577
 neighbor 10.4.4.2 description PEERING to VASI Global intf
 neighbor 10.4.4.2 activate
 exit-address-family
!
ip nat switchover replication http
ip nat pool att_pool 209.165.200.225 209.165.200.225 prefix-length 16
ip nat inside source list 4 pool att_pool redundancy 1 mapping-id 100 vrf VRFA overload
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip route 203.0.113.1 255.255.255.224 10.4.4.1
ip route 192.168.0.0 255.255.0.0 10.4.4.1
ip route 209.165.200.224 255.255.255.224 10.4.4.1
ip route vrf Mgmt-intf 209.165.200.1 255.255.255.224 172.16.0.0
!
ip prefix-list VRF_Pool seq 5 permit 209.165.200.0/27
ip prefix-list p1-adv-1 seq 5 permit 209.165.200.0/27
```

```
ip prefix-list p1-exist-1 seq 5 permit 203.0.113.193/27
logging esm config
access-list 4 permit 203.0.113.193 255.255.255.224
!
control-plane
line console 0
 stopbits 1
!
line vty 0 3
 login
!
line vty 4
 password lab
 login
!
end
```

Example: Configuring Asymmetric Routing with VRF

The following example shows how to configure asymmetric routing with virtual routing and forwarding (VRF) instances:

```
Device(config)# redundancy
Device(config-red)# mode sso
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name RG1
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# priority 100 failover threshold 40
Device(config-red-app-grp)# control GigabitEthernet 1/0/3 protocol 1
Device(config-red-app-grp)# data GigabitEthernet 1/0/3
Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 1/0/4
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# exit
Device(config-red-app)# exit
Device(config-red)# exit
!
Device(config)# interface TenGigabitEthernet 2/0/0
Device(config-if)# ip vrf forwarding vrf001
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# exit
!
Device(config)# interface TenGigabitEthernet 3/0/0
Device(config-if)# ip vrf forwarding vrf001
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# exit
!
Device(config-if)# ip nat pool pool-vrf001 209.165.201.1 209.165.201.30 prefix-length 24
Device(config-if)# ip nat inside source list 1 pool pool-vrf001 redundancy 1 mapping-id 1
vrf vrf001 match-in-vrf overload
Device(config-if)# end
```

Additional References for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

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
Firewall inter-chassis redundancy	"Configuring Firewall Stateful Inter-Chassis Redundancy" module
NAT inter-chassis redundancy	"Configuring Stateful Inter-Chassis Redundancy" module

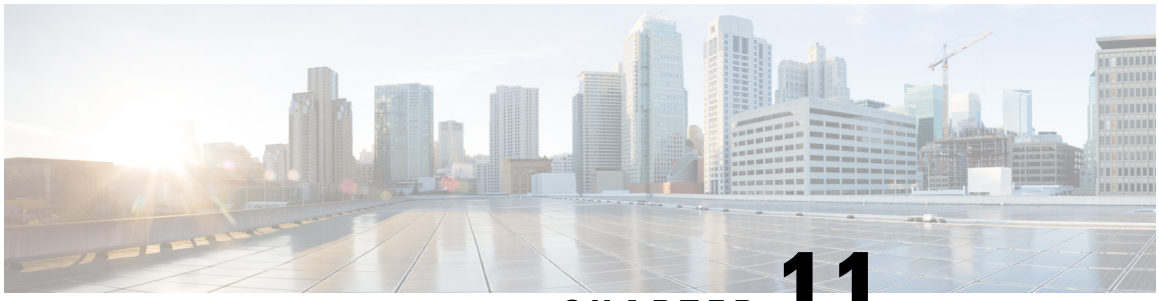
Technical Assistance

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 Firewall and NAT

Table 12: Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Feature Name	Releases	Feature Information
Asymmetric Routing Enhancements for NAT44	Cisco IOS XE Release 3.16S	The Asymmetric Routing Enhancements for NAT44 feature supports asymmetric routing with CGN, ALGs, VRF, VASI and MPLS. No commands were introduced or modified.
Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT	Cisco IOS XE Release 3.5S	The Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT 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**, **redundancy asymmetric-routing enable**.
VRF-Aware Interchassis Asymmetric Routing Support for Zone-Based Firewalls	Cisco IOS XE Release 3.14S	Zone-based firewalls support the VRF-Aware Interchassis Asymmetric Routing feature. This feature supports MPLS. There are no configuration changes for this feature. No commands were introduced or modified.
VRF-Aware Interchassis Asymmetric Routing Support for NAT	Cisco IOS XE Release 3.14S	NAT supports the VRF-Aware Interchassis Asymmetric Routing feature. This feature supports MPLS. There are no configuration changes for this feature. No commands were introduced or modified.



CHAPTER 11

Interchassis High Availability Support in IPv6 Zone-Based Firewalls

The Interchassis High Availability Support in IPv6 Zone-Based Firewalls feature supports asymmetric routing in firewalls that run IPv4 and IPv6 traffic at the same time. Asymmetric routing supports the forwarding of packets from a standby redundancy group to the 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.

This module provides an overview of asymmetric routing and describes how to configure asymmetric routing in IPv6 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.

Restrictions for Interchassis High Availability Support in IPv6 Zone-Based Firewalls

- Only IPv4 is supported at asymmetric-routing interlink interfaces.
- FTP64 application-level gateway (ALG) is not supported.
- LANs that use virtual IP addresses and virtual MAC (VMAC) addresses do not support asymmetric routing.
- Multiprotocol Label Switching (MPLS) and virtual routing and forwarding (VRF) instances are not supported because VRF ID mapping does not exist between active and standby Cisco ASR 1000 Series Aggregation Services Routers.

Information About Interchassis High Availability Support in IPv6 Zone-Based Firewalls

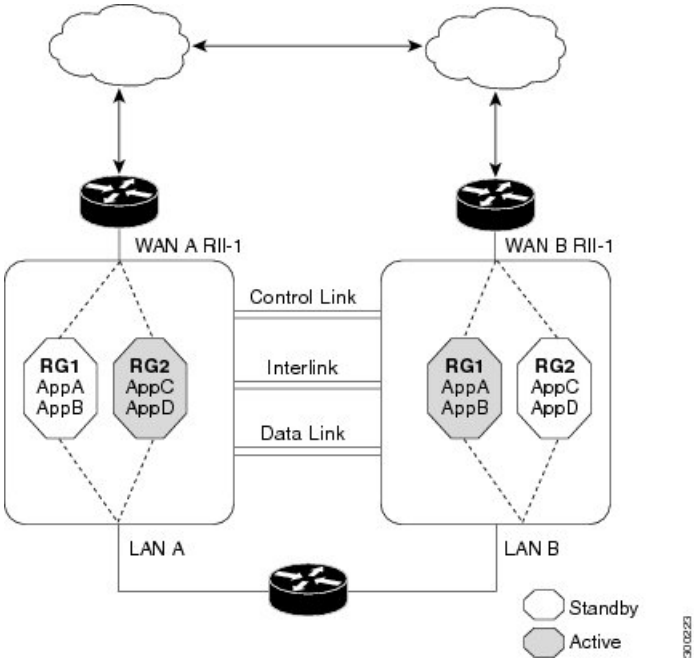
Asymmetric Routing Overview

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 17: 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.

Dual-Stack Firewalls

A dual-stack firewall is a firewall running IPv4 and IPv6 traffic at the same time. A dual-stack firewall can be configured in the following scenarios:

- One firewall zone running IPv4 traffic and another running IPv6 traffic.
- IPv4 and IPv6 coexist when deployed with stateful Network Address Translation 64 (NAT64). In this scenario, the traffic flows from IPv6 to IPv4 and vice versa.
- The same zone pair allows both IPv4 and IPv6 traffic.

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.

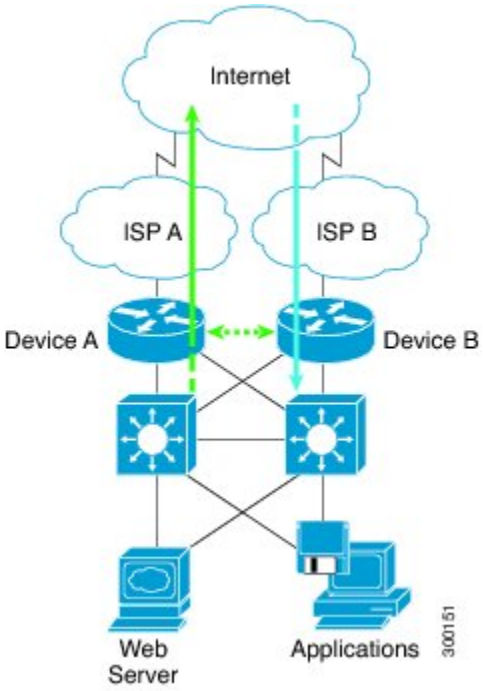
The firewall depends on bidirectional traffic to determine when a packet flow should be aged out and diverts all inspected packet flows to the active RG. Packet flows that have a pass policy and that include the same zone with no policy or a drop policy are not diverted.

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.

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 18: Asymmetric Routing in a WAN-LAN Topology



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 High Availability Support in IPv6 Zone-Based Firewalls

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

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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.

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.
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.

	Command or Action	Purpose
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.

Configuring an IPv6 Firewall

The steps to configure an IPv4 firewall and an IPv6 firewall are the same. To configure an IPv6 firewall, you must configure the class map in such a way that only an IPv6 address family is matched.

The **match protocol** command applies to both IPv4 and IPv6 traffic and can be included in either an IPv4 policy or an IPv6 policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf-definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **sessions maximum** *sessions*
9. **exit**
10. **ipv6 unicast-routing**
11. **ip port-map** *appl-name* **port** *port-num* **list** *list-name*
12. **ipv6 access-list** *access-list-name*
13. **permit ipv6 any any**
14. **exit**
15. **class-map type inspect match-all** *class-map-name*

16. **match access-group name** *access-group-name*
17. **match protocol** *protocol-name*
18. **exit**
19. **policy-map type inspect** *policy-map-name*
20. **class type inspect** *class-map-name*
21. **inspect** [*parameter-map-name*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf-definition** *vrf-name* **Example:** `Device(config)# vrf-definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect ipv6-param-map`	Enables a global inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
Step 8	**sessions maximum** *sessions* **Example:** `Device(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 9	**exit** **Example:**	Exits parameter-map type inspect configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-profile)# exit`	
Step 10	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 11	**ip port-map** *appl-name* **port** *port-num* **list** *list-name* **Example:** `Device(config)# ip port-map ftp port 8090 list ipv6-acl`	Establishes a port to application mapping (PAM) by using the IPv6 access control list (ACL).
Step 12	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 13	**permit ipv6 any any** **Example:** `Device(config-ipv6-acl)# permit ipv6 any any`	Sets permit conditions for an IPv6 access list.
Step 14	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 15	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ipv6-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.
Step 16	**match access-group name** *access-group-name* **Example:** `Device(config-cmap)# match access-group name ipv6-acl`	Configures the match criteria for a class map on the basis of the specified ACL.
Step 17	**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 18	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 19	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ipv6-policy`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.

	Command or Action	Purpose
Step 20	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ipv6-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 21	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect ipv6-param-map`	Enables stateful packet inspection.
Step 22	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring Zones and Zone Pairs for Asymmetric Routing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **ipv6 address** *ipv6-address/prefix-length*
12. **encapsulation dot1q** *vlan-id*
13. **zone-member security** *zone-name*
14. **end**
15. **show policy-map type inspect zone-pair sessions**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters 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 z1`	Creates a security zone and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** *zone-name* **Example:** `Device(config)# zone security z2`	Creates a security zone and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*] **Example:** `Device(config)# zone-pair security in-2-out source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy`	Attaches a policy map to a top-level policy map.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0.1`	Configures a subinterface and enters subinterface configuration mode.
Step 11	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-subif)# ipv6 address 2001:DB8:2222:7272::72/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface or a subinterface.
Step 12	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 2`	Sets the encapsulation method used by the interface.

	Command or Action	Purpose
Step 13	**zone-member security** *zone-name* **Example:** `Device(config-subif)# zone-member security z1`	Configures the interface as a zone member. • For the *zone-name* argument, you must configure one of the zones that you had configured 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 the 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 14	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.
Step 15	**show policy-map type inspect zone-pair sessions** **Example:** `Device# show policy-map type inspect zone-pair sessions`	Displays the stateful packet inspection sessions created because a policy map is applied on a specified zone pair. • The output of this command displays both IPv4 and IPv6 firewall sessions.

Configuration Examples for Interchassis High Availability Support in IPv6 Zone-Based Firewalls

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
```

Example: Configuring an IPv6 Firewall

```
Device# configure terminal
Device(config)# vrf-definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# parameter-map type inspect ipv6-param-map
Device(config-profile)# sessions maximum 10000
Device(config-profile)# exit
Device(config)# ipv6 unicast-routing
Device(config)# ip port-map ftp port 8090 list ipv6-acl
Device(config)# ipv6 access-list ipv6-acl
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# class-map type inspect match-all ipv6-class
Device(config-cmap)# match access-group name ipv6-acl
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect ipv6-policy
Device(config-pmap)# class type inspect ipv6-class
Device(config-pmap-c)# inspect ipv6-param-map
Device(config-pmap-c)# end
```

Example: Configuring Zones and Zone Pairs for Asymmetric Routing

```
Device# configure terminal
Device(config)# zone security z1
```

```
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security in-to-out source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0.1
Device(config-if)# ipv6 address 2001:DB8:2222:7272::72/64
Device(config-if)# encapsulation dot1q 2
Device(config-if)# zone member security z1
Device(config-if)# end
```

Additional References for Interchassis High Availability Support in IPv6 Zone-Based Firewalls

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

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

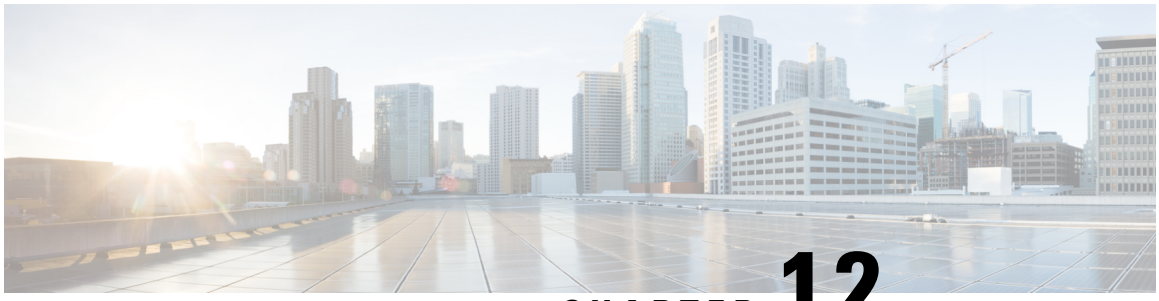
Feature Information for Interchassis High Availability Support in IPv6 Zone-Based Firewalls

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 13: Feature Information for Interchassis High Availability Support in IPv6 Zone-Based Firewalls

Feature Name	Releases	Feature Information
Interchassis High Availability Support in IPv6 Zone-Based Firewalls	Cisco IOS XE Release 3.8S	The Interchassis High Availability Support in IPv6 Zone-Based Firewalls feature supports asymmetric routing in firewalls that run IPv4 and IPv6 traffic at the same time. Asymmetric routing supports the forwarding of packets from a standby redundancy group to the 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. No commands were introduced or modified by this feature.



CHAPTER 12

Firewall Box to Box High Availability Support for Cisco CSR1000v Routers

The Firewall Box to Box High Availability Support on Cisco CSR1000v Routers feature enables you to configure pairs of routers to act as backup for each other. This feature 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 performing traffic forwarding services and maintaining a dynamic routing table.

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 Firewall Box-to-Box High Availability Support for Cisco CSR1000v Routers

- The interfaces attached to the firewall must have the same redundant interface identifier (RII).

- The active device and the standby device must have the same Cisco IOS XE Zone-Based Firewall configuration.
- The active device and the standby device must run on an identical version of the Cisco IOS XE software. The active device and the standby device must be connected through a switch.

Restrictions for Firewall Box-to-Box High Availability for Cisco CSR1000v Routers

- If the dual IOS daemon (IOSd) is configured, the device will not support the firewall box-to-box high availability configuration.

Information About Firewall Box to Box High Availability Support on Cisco CSR1000v Routers

How Firewall Box to Box High Availability Support on Cisco CSR1000v Works

You can configure pairs of routers to act as hot standbys for each other. This redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups. The figure below depicts the active-standby device scenario. It shows how the redundancy group is configured for a pair of routers that has one outgoing interface. The Redundancy Group Configuration—Two Outgoing Interfaces figure depicts the active-active device scenario shows how two redundancy groups are configured for a pair of routers that have two outgoing interfaces.

Note that in both cases, the redundant routers are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of the routers. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and to synchronize the stateful database for these applications.

Also, in both cases, the pairs of redundant interfaces are configured with the same unique ID number known as the RII.

Figure 19: Redundancy Group Configuration—Two Outgoing Interfaces

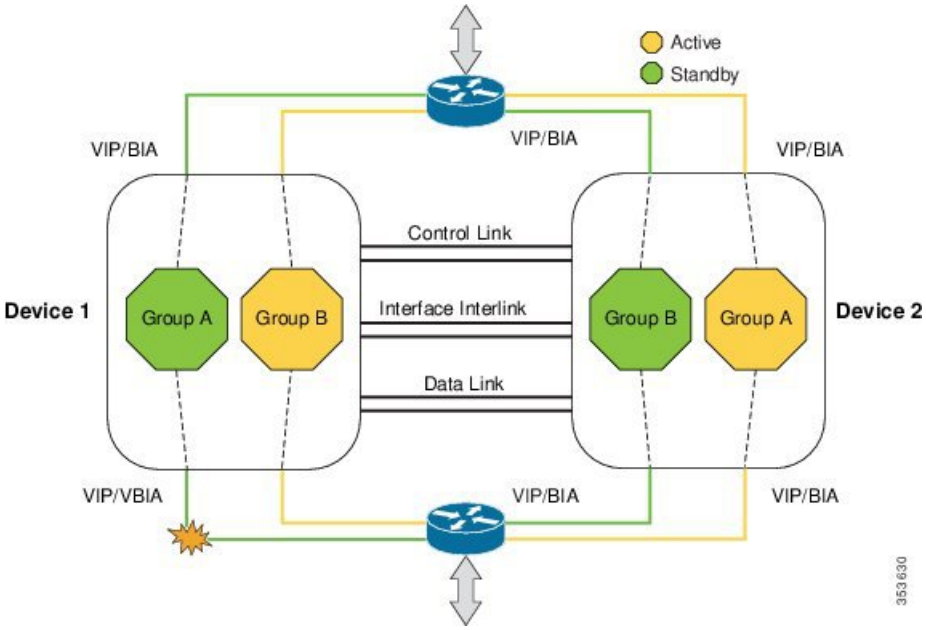
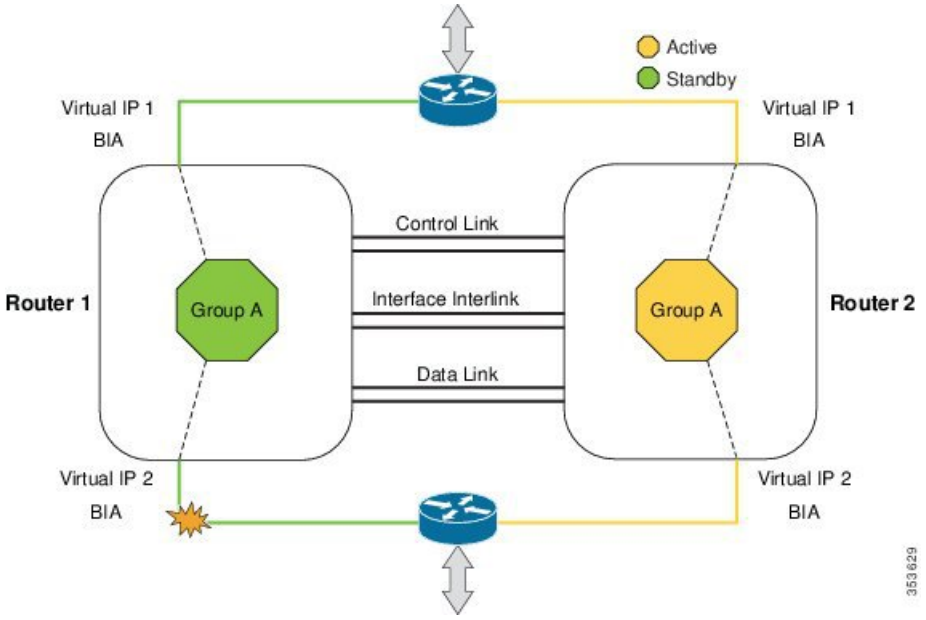
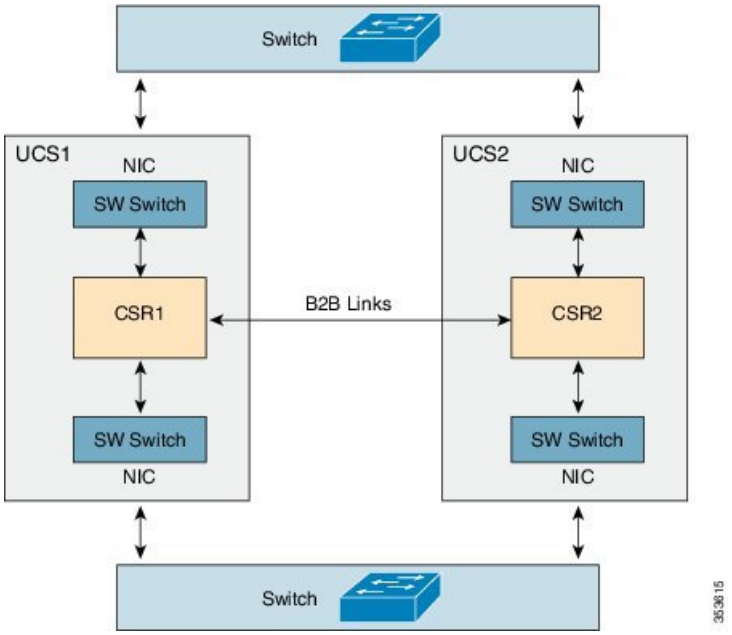


Figure 20: Redundancy Group Configuration



The following scenarios are examples of Box-to-Box High Availability deployment for Cisco CSR1000v routers:

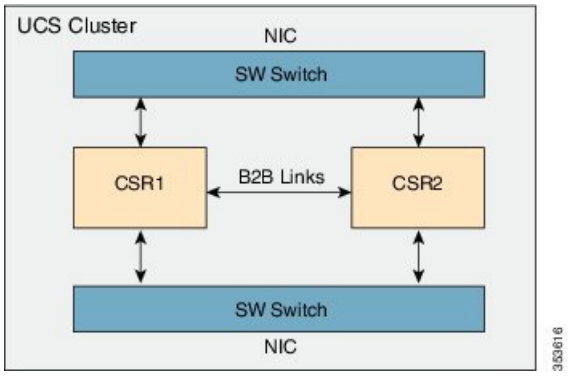
Figure 21: CSR1000v Box-to-Box High Availability on Two Independent Servers



In this deployment, two redundant Cisco CSR 1000v routers are in two independent UCS servers. The two Cisco Unified Computing System (UCS) servers can be in the same data center or two different data centers in different regions. We recommended that you configure two individual physical connections for box-to-box high availability data and control links. However, if the two dedicated physical links are not available, the box-to-box high availability data and control traffic can go through different LAN extension connections. Box-to-Box high availability parameters, such as heart beat period need to be adjusted to take into account the extended delay.

LAN interfaces of each Cisco CSR 1000v router are connected with UCS physical network interface card (NIC) interfaces through switches (for example, ESXi L2 SW). The two physical NICs on each UCS are connected to outside switch to form a box-to-box pair. Gratuitous Address Resolution Protocols (ARP) is sent from CSR LAN interfaces to reach physical switch and its Built-in Address (BIA).

Figure 22: CSR1000v Box-to-Box High Availability on Cluster Server



In the above deployment, NAT and Zone-Based Firewall (ZBFW) box-to-box high availability also works on UCS cluster setup. In this case, box-to-box control and data links go through virtual connections within the cluster. Switches (For example, ESXi L2 SW) are used to connect the 2 redundant Cisco CSR 1000v

routers to form a box-to-box high availability pair; LAN interfaces on two Cisco CSR 1000v routers are connected directly to the SW switches, and two physical NICs of the cluster UCS are connected with the SW switches to communicate outside the network.

Refer to the *Configuring Firewall Stateful Interchassis Redundancy* module for additional information on configurations and examples.

Configuration Example for Firewall Box-to-Box High Availability Support for Cisco CSR 1000v Routers

Example: Configuring Firewall Box-to-Box High Availability for Cisco CSR1000v Routers

The following examples shows how to configure a redundancy application group, a redundancy group protocol, Virtual IP Address and Redundant Interface Identifier, and control and data interfaces:

```
!Configures a redundancy application group
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 200 decrement 200
Device(config-red-app-grp)# exit

!Configures a redundancy group protocol
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9
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

! Configures a Virtual IP Address and Redundant Interface Identifier
Device# configure terminal
Device(config)# interface GigabitEthernet0/1/1
Device(conf-if)# redundancy rii 600
Device(config-if)# redundancy group 2 ip 10.2.3.4 exclusive decrement 200
Device(config)# redundancy
Device(config-red-app-grp)# data GigabitEthernet0/0/0
Device(config-red-app-grp)# control GigabitEthernet0/0/2 protocol 1
Device(config-red-app-grp)# end

!Configures control and data interfaces
Device# configure terminal
Device(config-red)# application redundancy
Device(config-red-app-grp)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/0
Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Device(config-red-app-grp)# end
```

Additional References for Firewall Box-to-Box High Availability for Cisco CSR1000v Routers

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
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
Firewall Stateful Interchassis Redundancy	• Configuring Firewall Stateful Interchassis Redundancy

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

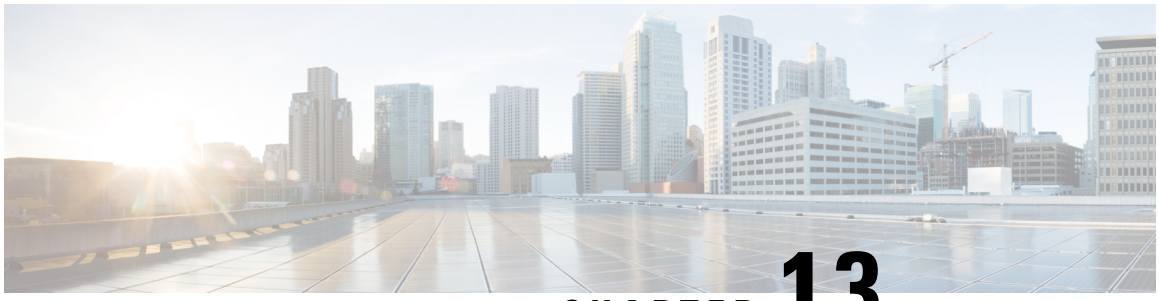
Feature Information for Firewall Box-to-Box High Availability for Cisco CSR1000v Routers

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 14: Feature Information for Firewall Stateful Interchassis Redundancy

Feature Name	Releases	Feature Information
Firewall Box-to-Box High Availability for Cisco CSR1000v Routers	Cisco IOS XE Release 3.14S	The Firewall Box-to-Box High Availability for Cisco CSR1000v Routers feature enables you to configure pairs of Cisco CSR1000v routers to act a backups for each other.



CHAPTER 13

Firewall Stateful Inspection of ICMP

The Firewall Stateful Inspection of ICMP feature categorizes Internet Control Management Protocol Version 4 (ICMPv4) messages as either malicious or benign. The firewall uses stateful inspection to *trust* benign ICMPv4 messages that are generated within a private network and permits the entry of associated ICMP replies into the network. The Firewall Stateful Inspection of ICMP feature helps network administrators to debug network issues by using ICMP so that intruders cannot enter the network.

This module provides an overview of the firewall stateful inspection of ICMPv4 messages and describes how to configure the firewall to inspect ICMPv4 messages.

Prerequisites for Firewall Stateful Inspection of ICMP

- You must configure the Cisco firewall before you can configure the Firewall Stateful Inspection of ICMP feature.
- The network must allow all ICMP traffic to pass through security appliance interfaces.
- Access rules must be configured for ICMP traffic that terminates at a security appliance interface.

Restrictions for Firewall Stateful Inspection of ICMP

This feature does not work with the UDP traceroute utility, in which UDP datagrams are sent instead of ICMP packets. UDP traceroute is the default for UNIX systems. For a UNIX host to generate ICMP traceroute packets that are inspected by the firewall, use the "-I" option with the **traceroute** command.

Information About Firewall Stateful Inspection of ICMP

Overview of the Firewall Stateful Inspection of ICMP

Internet Control Management Protocol (ICMP) is a network protocol that provides information about a network and reports errors in the network. Network administrators use ICMP to debug network connectivity issues. To guard against potential intruders using ICMP to discover the topology of a private network, ICMPv4 messages can be blocked from entering a private network; however, network administrators may then be unable to debug the network.

You can configure Cisco routers to use access control lists (ACLs) to either completely allow or deny ICMPv4 messages. When using ACLs for ICMPv4 messages, message *inspection* has precedence over the configured allow or deny actions.

ICMPv4 messages that use the IP protocol can be categorized into the following two types:

- Informational messages that utilize a simple request/reply mechanism.
- Error messages that indicate that some sort of error has occurred while delivering an IP packet.

Note To prevent ICMP attacks from using the Destination Unreachable error message, only one Destination Unreachable message is allowed per session by the firewall.

A host that is processing a UDP session that is traversing the firewall may generate an ICMP error packet with a Destination Unreachable message. In such cases, only one Destination Unreachable message is allowed through the firewall for that session.

The following ICMPv4 packet types are supported:

Table 15: ICMPv4 Packet Types

Packet Type	Name	Description
0	Echo Reply	Reply to an echo request (type 8).
3	Unreachable	Possible reply to any request.
8	Echo Request	Ping or a traceroute request.
11	Time Exceeded	Reply if the time-to-live (TTL) size of a packet is zero.
13	Timestamp Request	Request.
14	Timestamp Reply	Reply to a timestamp request (type 13).

ICMPv4 packet types 0 and 8 are used to ping a destination; the source sends out an Echo Request packet and the destination responds with an Echo Reply packet. Packet types 0, 8, and 11 are used for ICMPv4 traceroute (that is, Echo Request packets that are sent start with a TTL size of 1) and the TTL size is incremented for

each hop. Intermediate hops respond to the Echo Request packet with a Time Exceeded packet and the final destination responds with an Echo Reply packet.

If an ICMPv4 error packet is an embedded packet, the embedded packet is processed according to the protocol and the policy configured for the packet. For example, if the embedded packet is a TCP packet, and a drop action is configured for the packet, the packet is dropped even if ICMPv4 has configured a pass action.

The following scenario describes how ICMPv4 packets pass through the firewall:

1. An ICMPv4 packet arrives at the source interface. The firewall uses the source and destination addresses of the packet without any change for packet inspection. The firewall uses IP addresses (source and destination), the ICMP type, and the protocol for session key creation and lookup.
2. The packet passes the firewall inspection.
3. Return traffic comes from the destination interface and, based on the ICMPv4 message type, the firewall creates the session lookup key.
4. **a.** If the reply message is an informational message, the firewall uses the source and destination addresses from the packet without any change for packet inspection. Here, the destination port is the ICMPv4 message request type.
   b. If the reply message is an ICMPv4 error message, the firewall uses the payload packet present in the ICMP error packet to create the session key for session lookup.
5. If the firewall session lookup is successful, the packet passes the firewall inspection.

ICMP Inspection Checking

ICMP return packets are checked by the inspect code, and not by access control lists (ACLs). The inspect code tracks destination address from each outgoing packet and checks each return packet. For Echo Reply and Timestamp Reply packets, the return address is checked. For Unreachable and Time Exceeded packets, the intended destination address is extracted from the packet data and checked.

How to Configure Firewall Stateful Inspection of ICMP

Configuring Firewall Stateful Inspection of ICMP

Perform this task to configure the firewall stateful inspection of ICMP, which includes the following:

- A class map that matches the ICMP traffic.
- A policy map with the inspect action.
- Security zones and zone pairs (to attach a firewall policy map to the zone pair).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard*

4. **class-map type inspect** *class-map-name*
5. **match protocol** *protocol-name*
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class** *class-map-name*
9. **inspect**
10. **exit**
11. **exit**
12. **zone security** *zone-name*
13. **exit**
14. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
15. **service-policy type inspect** *policy-map-name*
16. **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	**access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* **Example:** `Device(config)# access-list 102 permit icmp 192.168.0.1 255.255.255.0 192.168.2.22 255.255.255.0`	Defines an extended IP access list.
Step 4	**class-map type inspect** *class-map-name* **Example:** `Device(config)# class-map type inspect c1`	Defines the class on which an action is to be performed and enters QoS class-map configuration mode.
Step 5	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol icmp`	Configures a match criterion for a class map on the basis of the specified protocol.
Step 6	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p1`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 8	**class** *class-map-name* **Example:** `Device(config-pmap)# class c1`	Defines the class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 9	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 10	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 11	**exit** **Example:** `Device(config-pmap)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 12	**zone security** *zone-name* **Example:** `Device(config)# zone security z1`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source zone and a destination zone. • In a zone pair, you can use the default zone as either the source or the destination zone.
Step 13	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 14	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security inout source z1 destination z2`	Creates a zone pair to which interfaces can be assigned and enters security zone-pair configuration mode.
Step 15	**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.

	Command or Action	Purpose
Step 16	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and enters privileged EXEC mode.

Verifying Firewall Stateful Inspection of ICMP

You can use the following **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **show ip access-lists**
3. **show policy-map type inspect** *policy-map-name*
4. **show policy-map type inspect zone-pair** *zone-pair-name*
5. **show zone security** *zone-name*
6. **show zone-pair security** [**source** *source-zone* **destination** *destination-zone*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip access-lists**

Example:

```
Device# show ip access-lists
```

Displays information about the specified policy map.

Step 3 **show policy-map type inspect** *policy-map-name*

Example:

```
Device# show policy-map type inspect p1
```

Displays information about the specified policy map.

Step 4 **show policy-map type inspect zone-pair** *zone-pair-name*

Example:

```
Device# show policy-map type inspect zone-pair inout
```

Displays the runtime inspect type policy-map statistics for the zone pair.

Step 5 **show zone security** *zone-name*

Example:

```
Device# show zone security z1
```

Displays zone security information.

Step 6 **show zone-pair security** [**source** *source-zone* **destination** *destination-zone*]

Example:

```
Device# show zone-pair security source z1 destination z2
```

Displays source and destination zones and the policy attached to the zone pair.

Example:

The following sample output from the **show ip access-lists** command shows how ACLs are created for an ICMP session for which only ping packets were issued from the host:

```
Device# show ip access-lists

 Extended IP access list 102
     permit icmp any host 192.168.133.3 time-exceeded
     permit icmp any host 192.168.133.3 unreachable
     permit icmp any host 192.168.133.3 timestamp-reply
     permit icmp any host 192.168.133.3 echo-reply (4 matches)
```

The following is sample output from the **show policy-map type inspect p1** command:

```
Device# show policy-map type inspect p1

 Policy Map type inspect p1
  Class c1
   Inspect
```

The following is sample output from the **show policy-map type inspect zone-pair inout** command:

```
Device# show policy-map type inspect zone-pair inout

 Zone-pair: inout
  Service-policy : p1
   Class-map: c1 (match-all)
    Match: protocol icmp
    Inspect
     Session creations since subsystem startup or last reset 0
     Current session counts (estab/half-open/terminating) [0:0:0]
     Maxever session counts (estab/half-open/terminating) [0:0:0]
     Last session created never
     Last statistic reset never
     Last session creation rate 0
     half-open session total 0
   Class-map: class-default (match-any)
    Match: any
    Drop
     0 packets, 0 bytes
```

The following is sample output from the **show zone security** command:

```
Device# show zone security
```

```
zone self
Description: System defined zone
```

The following is sample output from the **show zone-pair security** command:

```
Device# show zone-pair security source z1 destination z2

zone-pair name inout
  Source-Zone z1  Destination-Zone z2
  service-policy p1
```

Configuration Examples for Firewall Stateful Inspection of ICMP

Example: Configuring Firewall Stateful Inspection of ICMP

```
Device# configure terminal
Device(config)# access-list 102 permit icmp 192.168.0.1 255.255.255.0 192.168.2.22
255.255.255.0
Device(config)# class-map type inspect c1
Device(config-cmap)# match protocol icmp
Device(config-cmap)# exit
Device(config)# policy-map type inspect p1
Device(config-pmap)# class c1
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# zone security z1
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security inout source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect p1
Device(config-sec-zone-pair)# end
```

Additional References for Firewall Stateful Inspection of ICMP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
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 & RFCs

Standard/RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 950	*Internet Standard Subnetting Procedure*
RFC 1700	*Assigned Numbers*

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

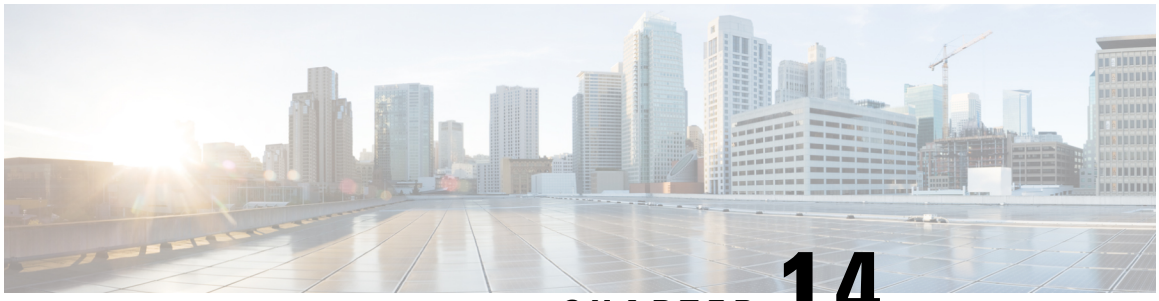
Feature Information for Firewall Stateful Inspection of ICMP

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 16: Feature Information for Firewall Stateful Inspection of ICMP

Feature Name	Releases	Feature Information
Firewall Stateful Inspection of ICMP	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2S	The Firewall Stateful Inspection of ICMP feature categorizes ICMPv4 messages as either malicious or benign. The firewall uses stateful inspection to *trust* benign ICMP messages that are generated within a private network and permits the entry of associated ICMP replies.



CHAPTER 14

Application Aware Firewall

This document describes how Zone Based FireWall policy is defined based on the applications that NBAR can detect and make Zone Based FireWall application aware. The Application FireWall inspects the traffic and blocks traffic based on applications, category, application-family or application-group. This application aware firewall feature provides the following benefits:

- Application visibility and granular control
- Classification of 1400+ layer 7 applications
- Allows or blocks traffic by application, category, application-family or application-group

Feature Information for Application Aware 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.

Feature Name	Releases	Feature Information
Application Aware Zone-based FW	Cisco IOS XE Fuji 16.9.1	This document describes how Zone Based FireWall policy is defined based on the applications that NBAR can detect and make Zone Based FireWall application aware. The Application FireWall inspects the traffic and blocks traffic based on applications, category, application-family or application-group. The following commands were introduced or modified: **show class-map***avc-classmap-name* **show policy-map type inspect zone-pair** **show policy-map type inspect zone-pair sessions** **show policy-map type inspect avc** **show platform hardware qfpactive feature firewall drop**

Information About Application Awareness on Zone-Based FW

Prerequisites for Application Aware Firewall

- Ensure that traffic is matched to the Layer3/Layer4 inspect class map. If the traffic does not match the firewall inspection, the AVC policy fails to see the traffic.
- Inspect DNS in the same class-map where the AVC service-policy is applied.

Restrictions on Application Aware Zone-Based FW

- No support for traffic to self-zone.
- The AVC inspect policy should allow all and only deny certain application because many applications are interdependent and therefore allowing one application while denying all others do not work all the time.
- Each application class-map can have upto 16 filters (each match is considered a filter).
- The AVC policy-map can have upto 32 class-maps (including class-default).
- You cannot configure **match protocol attribute application-family** or **match protocol attribute application-group** if you specify the category using the **match protocol attribute category** command.

Before you configure class-map and policy-map, use the **parameter-map type inspect** configure the parameter-map type to log dropped packets:

```
Device (config)# parameter-map type inspect
Device (config-map)# log dropped-packets
```

Policies Based on Network Layers L3/L4

Zone-based Firewall uses policies based on network layers L3/L4, for example, class maps are based on ACL and L4 protocolsTCP/UDP/ICMP or L7 protocolsFTP and SIP. Policies that are defined using the L7 protocol utilize the protocol's destination port to classify the packet. ZBF lacks application visibility, it supports FTP inspection through the FTP ALG, and only identifies the protocols that are based on port 21.

Note If an FTP control flow is opened on some random port, zone-based firewall cannot identify the application.

How to Configure NBAR Based Application Awareness on ZBFW

Configure Layer 4 Zone-Based Firewall

```
Device(config-profile)#class-map type inspect match-any cm1
Device(config-cmap)#match protocol http
Device(config-cmap)#match protocol https
Device(config-cmap)#match protocol dns
Device(config-cmap)#match protocol tcp
Device(config-cmap)#match protocol udp
Device(config-cmap)#match protocol icmp
Device(config-cmap)#exit
Device(config)#class-map match-any nbar-class1
Device(config-cmap)#match protocol yahoo-mail
Device(config-cmap)#match protocol amazon
Device(config-cmap)#match protocol attribute category consumer-internet
Device(config-cmap)#exit
```

L7 Service Policy for Application Aware Firewall

Procedure

	Command or Action	Purpose

Step 1	Configure the class-map for inspection.
	Example:
	`class-map type inspect match-any cm1`
	` match protocol http`
	` match protocol https`
	` match protocol dns`
	` match protocol tcp`
	` match protocol udp`
	` match protocol icmp`
		Defines the protocols and category using the **class-map type inspect** and **match protocol** commands.

Step 2	Define the action, in this case the AVC, using the application firewall policy.
	Example:
	`policy-map type inspect avc nbar-policy1`
	`class nbar-class1`
		Uses the **deny** command to refuse the remote network management protocols listed in the `nbar-class1`class map.

	Command or Action	Purpose
	deny class class-default allow	
Step 3	Log the dropped packets using the application firewall policy. **Example:** policy-map type inspect pm1 class type inspect cm1 inspect service-policy avc nbar-policy1 class class-default drop log Traffic from amazon, in nbar-class1, is denied by the policy. For example, a dropped packet is shown in the following drop log message: Oct 17 12:44:08.101: %IOSXE-6-PLATFORM: R0/0: cpp_cp: QFP:0.0 Thread:000 TS:00000002517650404876 %FW-6-DROP_PKT: Dropping dns/amazon pkt from GigabitEthernet3 171.70.168.183:53 => 171.10.1.101:50877(target:class) -(in_to_out:cm1) due to AVC Policy drop:classify result with ip ident 65434	

What to do next

Add the **ip nbar protocol-discovery ipv4** command on the ingress interface. Then use the **show ip nbar protocol-discovery interface** [**intf-name**] command to see the application classification.

Example: Application Aware Show Commands

In this example, the **show policy-map type inspect zone-pair** command shows the policy map statistics and other information including information about the sessions existing on a specified zone pair. The line following `Class-map: nbar-class1 (match-any)` includes the packet counter value (`7 packets`), which increases whenever traffic matches the nbar-class1 class.

```
Device# show policy-map type inspect zone-pair

Zone-pair: in_to_out
Service-policy inspect : pm1

Class-map: cm1 (match-any)
Match: protocol http
Match: protocol https
Match: protocol dns
Match: protocol tcp
Match: protocol udp
Match: protocol icmp
Inspect
Packet inspection statistics [process switch:fast switch]
tcp packets: [0:485]
dns packets: [0:51]
```

```
Session creations since subsystem startup or last reset 21
Current session counts (estab/half-open/terminating) [13:0:0]
Maxever session counts (estab/half-open/terminating) [13:2:0]
Last session created 00:00:00
Last statistic reset 00:00:19
Last session creation rate 151
Last half-open session total 0

Service-policy inspect avc : nbar-policy1

Class-map: nbar-class1 (match-any)
7 packets, 1449 bytes
30 second offered rate 1000 bps, drop rate 0000 bps
Match: protocol amazon
Match: protocol yahoo-mail
Match: protocol attribute category consumer-internet
Deny

Class-map: class-default (match-any)
211 packets, 94091 bytes
30 second offered rate 27000 bps, drop rate 0000 bps
Match: any
Allow

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

```
Device# show platform hardware qfp active feature firewall drop
-------------------------------------------------------------------------------
Drop Reason                                                            Packets
-------------------------------------------------------------------------------
AVC Policy drop:classify result                                             38
```

```
Device# show platform hardware qfp active feature firewal datapath scb
[s=session  i=imprecise channel c=control channel  d=data channel A/D=appfw action allow/deny]
Session ID:0x0000DA5B  171.10.1.101 64204 171.70.168.183 53 proto 17 (0:0) (1456:0xd000208)
 [scA]
Session ID:0x0000DA18  171.10.1.101 58836 74.125.199.103 443 proto 6 (0:0) (1456:0xd000208)
 [sdA]
Session ID:0x0000DA5A  171.10.1.101 64206 8.8.8.8 53 proto 17 (0:0) (0:0xd000001) [sc]
Session ID:0x0000DA11  171.10.1.101 58833 74.125.199.84 443 proto 6 (0:0) (1440:0xd000210)
 [sdA]
Session ID:0x0000DA57  171.10.1.101 64205 173.36.131.10 53 proto 17 (0:0) (1761:0xd00033f)
 [scD]
Session ID:0x0000DA2C  171.10.1.101 58839 74.125.199.94 443 proto 6 (0:0) (1456:0xd000208)
 [sdA]
Session ID:0x0000DA59  171.10.1.101 64203 173.36.131.10 53 proto 17 (0:0) (1761:0xd00033f)
 [scD]
Session ID:0x0000DA0B  171.10.1.101 58831 74.125.199.94 443 proto 6 (0:0) (1456:0xd000208)
 [sdA]
Session ID:0x0000DA5C  171.10.1.101 64207 8.8.4.4 53 proto 17 (0:0) (0:0xd000001) [sc]
Session ID:0x0000DA58  171.10.1.101 64203 171.70.168.183 53 proto 17 (0:0) (1761:0xd00033f)
 [scD]
```

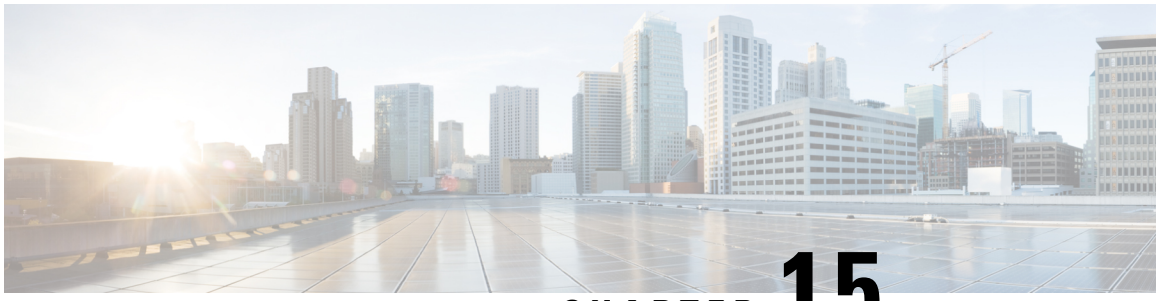
Additional References for Firewall Stateful Interchassis Redundancy

Related Documents

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

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



CHAPTER 15

Firewall Support of Skinny Client Control Protocol

The Firewall Support of Skinny Client Control Protocol feature enables the Cisco IOS XE firewall to support VoIP and the Skinny Client Control Protocol (SCCP). Cisco IP phones use the SCCP to connect with and register to Cisco Unified Communications Manager. To be able to configure Cisco IOS XE firewall between the IP phone and Cisco Unified Communications Manager in a scalable environment, the firewall needs to be able to detect SCCP and understand the information passed within the messages. With the Firewall Support of Skinny Client Control Protocol feature, the firewall inspects Skinny control packets that are exchanged between Skinny clients (such as IP Phones) and the Cisco Unified Communications Manager and configures the router to enable Skinny data channels to traverse through the router. This 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 Firewall Support of Skinny Client Control Protocol

- Your system must be running Cisco IOS XE Release 2.1 or a later release.
- You must enable the firewall for the SCCP application-level gateway (ALG) to work.
- You must enable the TFTP ALG for SCCP to work because IP phones that use Skinny need the TFTP configuration file from the Cisco Unified Communications Manager.

Restrictions for Firewall Support of Skinny Client Control Protocol

- IPv6 address inspection and translation is not supported.
- TCP segmentation is not supported.

Information About Firewall Support of Skinny Client Control Protocol

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

SCCP Inspection Overview

SCCP inspection enables voice communication between two SCCP clients by using the Cisco Unified Communications Manager. The Cisco Unified Communications Manager uses the TCP port 2000 (the default

SCCP port) to provide services to SCCP clients. Initially, the SCCP client connects to the primary Cisco Unified Communications Manager by establishing a TCP connection and, if available, connects to a secondary Cisco Unified Communications Manager. After the TCP connection is established, the SCCP client registers with the primary Cisco Unified Communications Manager, which is used as the controlling Cisco Unified Communications Manager until it reboots or a keepalive failure occurs. Thus, the TCP connection between the SCCP client and the Cisco Unified Communications Manager exists forever and is used to establish calls coming to or from the client. If a TCP connection fails, the secondary Cisco Unified Communications Manager is used. All data channels established with the initial Cisco Unified Communications Manager remain active and will be closed after the call ends.

The SCCP protocol inspects the locally generated or terminated SCCP 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 table below lists the set of messages that are necessary for the data sessions to open and close. SCCP inspection will examine the data sessions that are used for opening and closing the access list pinholes.

Table 17: SCCP Data Session Messages

Skinny Inspection Message	Description
CloseReceiveChannel	Indicates that the call should be aborted. Any intermediate sessions created by the firewall and NAT have to be cleaned up when this message is received.
OpenReceiveChannelACK	Indicates that the phone is acknowledging the OpenReceiveChannel message that it received from the Cisco Unified Communications Manager.
StartMediaTransmission	Contains the Realtime Transport Protocol (RTP) information of the phone that is the source or destination of the call. The message contains the IP address, the RTP port that the other phone is listening on, and the Call ID that uniquely identifies the call.
StopMediaTransmission	Indicates that the call has ended. Sessions can be cleaned up after receiving this message.
StationCloseReceiveChannel	Instructs the Skinny client (on the basis of the information in this message) to close the receiving channel.
StationOpenMultiMediaReceiveChannelAck	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.
StationOpenReceiveChannelAck	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.
StationStartMediaTransmission	Contains the IP address and port information of the remote Skinny client.

Skinny Inspection Message	Description
StationStartMultiMediaTransmit	Indicates that the Cisco Unified Communications Manager received an OpenLogicalChannelAck message for the video or the data channel.
StationStopMediaTransmission	Instructs the Skinny client (on the basis of the information in this message) to stop transmitting voice traffic.
StationStopSessionTransmission	Instructs the Skinny client (on the basis of the information in this message) to end the specified session.

ALG--SCCP Version 17 Support

The ALG—SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP Version 17 packets. Cisco Unified Communications Manager 7.0 and the IP phones that use Cisco Unified Communications Manager 7.0 support only SCCP Version 17 messages. The format of SCCP changed from Version 17 to support IPv6. The SCCP ALG checks for the SCCP version in the prefix of a message before parsing it according to the version. The SCCP message version is extracted from the message header and if it is greater than Version 17, the message is parsed by using the Version 17 format and the IPv4 address and port information is extracted. The SCCP ALG supports the inspection and translation of IPv4 address information in SCCP messages.

Note IPv6 address inspection and translation are not supported.

The IP address format of the following SCCP ALG-handled messages changed in Version 17:

- StationOpenMultiMediaReceiveChannelAck
- StationOpenReceiveChannelAckMessage
- StationRegisterMessage
- StationStartMediaTransmissionAckMessage
- StationStartMultiMediaTransmissionAckMessage
- StationStartMediaTransmissionMessage
- StationStartMultiMediaTransmissionMessage

How to Configure Firewall Support of Skinny Client Control Protocol

Configuring a Skinny Class Map and Policy Map

When you enable SCCP (through the **match protocol** command) in a firewall configuration, you must enable TFTP (through the **match protocol** command); otherwise, the IP phones that use SCCP cannot communicate with the Cisco Unified Communications Manager. SCCP enables voice communication between two Skinny clients through the use of a Cisco Unified Communications Manager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **match protocol** *protocol-name*
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class type inspect** *class-map-name*
9. **inspect**
10. **exit**
11. **class class-default**
12. **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-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any cmap1`	Creates an inspect type class map and enters class map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol skinny`	Configures the match criterion for a Skinny class map.

	Command or Action	Purpose
Step 5	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol tftp`	Configures the match criterion for a TFTP class map.
Step 6	**exit** **Example:** `Router(config-cmap)# exit`	Exits class map configuration mode.
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect pmap1`	Creates an inspect type policy map and enters policy map configuration mode.
Step 8	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect cmap1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 9	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 10	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode and enters policy map configuration mode.
Step 11	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies that these policy map settings apply to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 12	**end** **Example:** `Router(config-pmap)# end`	Exits policy map configuration mode and enters privileged EXEC mode.

Configuring a Zone Pair and Attaching an SCCP Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}

6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
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	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned 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.
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security zone2`	Creates a security zone to which interfaces can be assigned 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-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]] **Example:** `Router(config)# zone-pair security in-out source zone1 destination zone2`	Creates a zone pair and enters security zone pair configuration mode. **Note** To apply a policy, you must configure a zone pair.

	Command or Action	Purpose
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect pmap1`	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 9	**exit** **Example:** `Router(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 13	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for Firewall Support of Skinny Control Protocol

Example: Configuring an SCCP Class Map and a Policy Map

```
Router# configure terminal
Router(config)# class-map type inspect match-any cmap1
Router(config-cmap)# match protocol skinny
Router(config-cmap)# match protocol tftp
Router(config-cmap)# exit
Router(config)# policy-map type inspect pmap1
Router(config-pmap)# class type inspect cmap1
Router(config-pmap-c)# inspect
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap)# end
```

Example: Configuring a Zone Pair and Attaching an SCCP Policy Map

```
Router# configure terminal
Router(config)# zone security zone1
Router(config-sec-zone)# exit
Router(config)# zone security zone2
Router(config-sec-zone)# exit
Router(config)# zone-pair security in-out source zone1 destination zone2
Router(config-sec-zone-pair)# service-policy type inspect pmap1
Router(config-sec-zone-pair)# exit
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# zone-member security zone1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/1/1
Router(config-if)# zone-member security zone2
Router(config-if)# end
```

Additional References for Firewall Support of Skinny Client Control Protocol

Related Documents

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

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

Technical Assistance

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

Feature Information for Firewall Support for Skinny Client Control Protocol

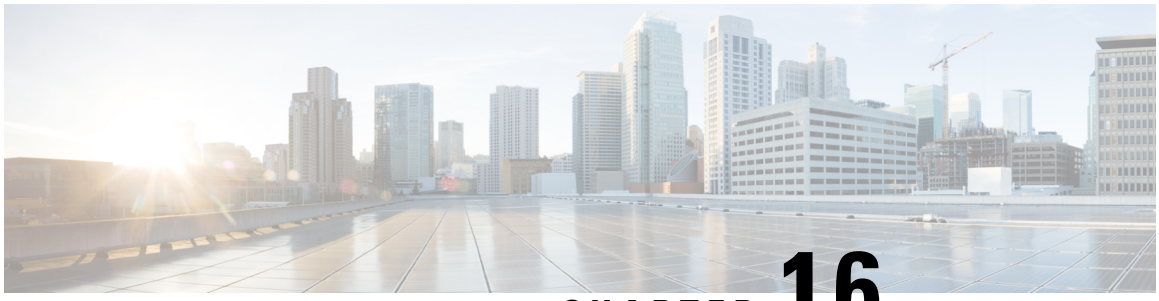
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

Feature Name	Releases	Feature Information
ALG—SCCP V17 Support	Cisco IOS XE Release 3.5S	The ALG—SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP version 17 packets. The SCCP format has changed from version 17 to support IPv6.

Feature Name	Releases	Feature Information
Firewall—SCCP Video ALG Support	Cisco IOS XE Release 2.4	SCCP enables voice communication between two Skinny clients through the use of a Cisco Unified Communications Manager. This feature enables Cisco firewalls to inspect Skinny control packets that are exchanged between a Skinny client and the Cisco Unified Communications Manager. The following command was modified: **match protocol**.
Firewall Support for Skinny Client Control Protocol	Cisco IOS XE Release 2.1	The Firewall Support of Skinny Client Control Protocol feature enables the Cisco IOS XE firewall to support VoIP and SCCP. Cisco IP phones use the SCCP to connect with and register to Cisco Unified Communications Manager. To be able to configure Cisco IOS XE firewall between the IP phone and Cisco Unified Communications Manager in a scalable environment, the firewall needs to be able to detect SCCP and understand the information passed within the messages. With the Firewall Support of Skinny Client Control Protocol feature, the firewall inspects Skinny control packets that are exchanged between Skinny clients (such as IP Phones) and the Cisco Unified Communications Manager and configures the router to enable Skinny data channels to traverse through the router. This feature extends the support of SCCP to accommodate video channels..



CHAPTER 16

Configuring the VRF-Aware Software Infrastructure

The VRF-Aware Software Infrastructure feature allows you to apply services such as, access control lists (ACLs), Network Address Translation (NAT), policing, and zone-based firewalls, to traffic that flows across two different virtual routing and forwarding (VRF) instances. VRF-Aware Software Infrastructure (VASI) interfaces support the redundancy of Route Processors (RPs) and Forwarding Processors (FPs), IPsec, and IPv4 and IPv6 unicast and multicast traffic.

This module describes how to configure VASI interfaces.

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 Configuring the VRF-Aware Software Infrastructure

- Multiprotocol Label Switching (MPLS) traffic over VRF-Aware Software Infrastructure (VASI) interfaces is not supported.

- VASI interfaces do not support the attachment of queue-based features. The following commands are not supported on Modular QoS CLI (MQC) policies that are attached to VASI interfaces:
  - **bandwidth (policy-map class)**
  - **fair-queue**
  - **priority**
  - **queue-limit**
  - **random-detect**
  - **shape**
- VASI 2000 pairs are not supported on Open Shortest Path First (OSPF).
- VASI is not supported because Multicast First Hop and Multicast punt packets on VASI interface are not supported.
- Web Cache Communication Protocol (WCCP) is not supported.

Information About Configuring the VRF-Aware Software Infrastructure

VASI Overview

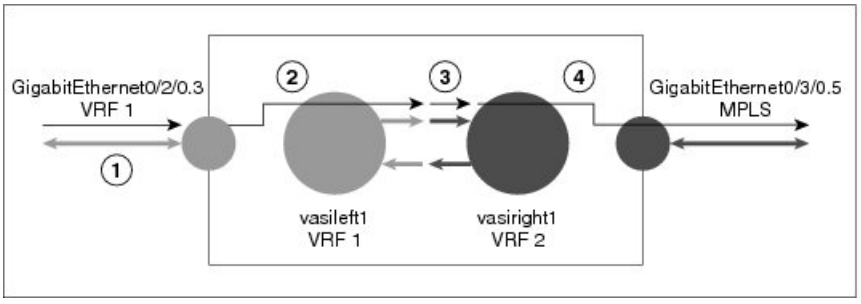
VRF-Aware Software Infrastructure (VASI) provides the ability to apply services such as, a firewall, GETVPN, IPsec, and Network Address Translation (NAT), to traffic that flows across different virtual routing and forwarding (VRF) instances. VASI is implemented by using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF instance. The VASI virtual interface is the next-hop interface for any packet that needs to be switched between these two VRF instances. VASI interfaces provide the framework to configure a firewall or NAT between VRF instances.

Each interface pair is associated with two different VRF instances. The pairing is done automatically based on the two interface indexes such that the vasileft interface is automatically paired to the vasiright interface. For example, in the figure below, vasileft1 and vasiright1 are automatically paired, and a packet entering vasileft1 is internally handed over to vasiright1.

On VASI interfaces, you can configure either static routing or dynamic routing with Internal Border Gateway Protocol (IBGP), Enhanced Interior Gateway Routing Protocol (EIGRP), or Open Shortest Path First (OSPF).

The following figure shows an inter-VRF VASI configuration on the same device.

Figure 23: Inter-VRF VASI Configuration



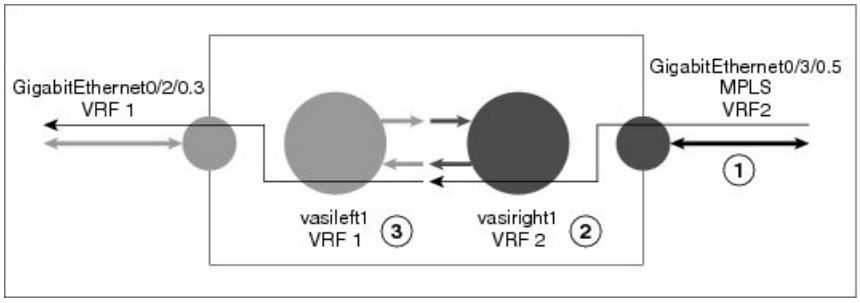
When an inter-VRF VASI is configured on the same device, the packet flow happens in the following order:

1. A packet enters the physical interface that belongs to VRF 1 (Gigabit Ethernet 0/2/0.3).
2. Before forwarding the packet, a forwarding lookup is done in the VRF 1 routing table. Vasileft1 is chosen as the next hop, and the Time to Live (TTL) value is decremented from the packet. Usually, the forwarding address is selected on the basis of the default route in the VRF. However, the forwarding address can also be a static route or a learned route. The packet is sent to the egress path of vasileft1 and then automatically sent to the vasiright1 ingress path.
3. When the packet enters vasiright1, a forwarding lookup is done in the VRF 2 routing table, and the TTL is decremented again (second time for this packet).
4. VRF 2 forwards the packet to the physical interface, Gigabit Ethernet 0/3/0.5.

The following figure shows how VASI works in a Multiprotocol Label Switching (MPLS) VPN configuration.

Note In the following figure, MPLS is enabled on the Gigabit Ethernet interface, but MPLS traffic is not supported across VASI pairs.

Figure 24: VASI with an MPLS VPN Configuration



When VASI is configured with a Multiprotocol Label Switching (MPLS) VPN, the packet flow happens in the following order:

1. A packet arrives on the MPLS interface with a VPN label.
2. The VPN label is stripped from the packet, a forwarding lookup is done within VRF 2, and the packet is forwarded to vasiright1. The TTL value is decremented from the packet.
3. The packet enters vasileft1 on the ingress path, and another forwarding lookup is done in VRF 1. The packet is sent to the egress physical interface in VRF1 (Gigabit Ethernet 0/2/0.3). The TTL is again decremented from the packet.

Multicast and Multicast VPN on VASI

VRF-Aware Service Infrastructure (VASI) applies services like the zone-based firewall, Network Address Translation (NAT), and IPsec to traffic that travels across different virtual routing and forwarding (VRF) instances. The Multicast and MVPN on VASI feature supports IPv4 and IPv6 multicast and multicast VPN (MVPN) on VASI interfaces. This feature is independent of the multicast modes (sparse, source-specific multicast [SSM] and so on) configured at the customer site and also independent of the MVPN mode—generic

routing encapsulation (GRE)-based or Multicast Label Distribution Protocol (MLDP)-based—in the core network.

Multicast reduces traffic in a network by simultaneously delivering a single stream of information to potentially thousands of recipients. Multicast delivers source traffic from an application to multiple receivers without burdening the source or receivers and uses a minimum of network bandwidth. Multicast VPN (MVPN) provides the ability to support multicast over Layer 3 VPNs.

VASI is implemented using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF. VASI virtual interface is the next hop interface for any packet that needs to be switched between these two VRFs. VASI interfaces are virtual interfaces and you can configure IP address and other services like other logical interfaces. You need to enable multicast on VASI interface pairs for this feature to work.

How to Configure the VRF-Aware Software Infrastructure

Configuring a VASI Interface Pair

To configure a VRF-Aware Software Infrastructure (VASI) interface pair, you must configure the **interface vasileft** command on one interface and the **interface vasiright** command on the second interface. The interface numbers must be identical to pair vasileft with vasiright. You can configure a virtual routing and forwarding (VRF) instance on any VASI interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *table-name*
5. **ip address** {*ip-address mask* [**secondary**] | **pool** *pool-name*}
6. **exit**
7. **ip route** [**vrf** *vrf-name*] *destination-prefix destination-prefix-mask interface-type interface-number*
8. **interface** *type number*
9. **vrf forwarding** *table-name*
10. **ip address** {*ip-address mask* [**secondary**] | **pool** *pool-name*}
11. **exit**
12. **ip route** [**vrf** *vrf-name*] *destination-prefix destination-prefix-mask interface-type interface-number*
13. **end**

DETAILED STEPS

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

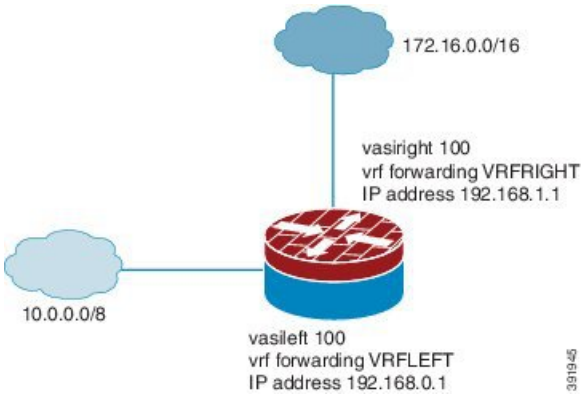
	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface vasileft 100`	Configures a VASI interface and enters interface configuration mode. • In this example, the vasileft interface is configured.
Step 4	**vrf forwarding** *table-name* **Example:** `Device(config-if)# vrf forwarding VRFLEFT`	Configures a VRF table. **Note** You can configure VRF forwarding on any VASI interface. You need not configure VRF instances on both VASI interfaces.
Step 5	**ip address** {*ip-address mask* [**secondary**] | **pool** *pool-name*} **Example:** `Device(config-if)# ip address 192.168.0.1 255.255.255.0`	Configures a primary or secondary IP address for an interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**ip route** [**vrf** *vrf-name*] *destination-prefix destination-prefix-mask interface-type interface-number* **Example:** `Device(config)# ip route vrf VRFLEFT 172.16.0.0 255.255.0.0 VASILEFT 100`	Establishes a static route for a VRF instance and a VASI interface. **Note** To add an IP route for a VRF instance, you must specify the **vrf** keyword.
Step 8	**interface** *type number* **Example:** `Device(config)# interface vasiright 100`	Configures a VASI interface and enters interface configuration mode. • In this example, the vasiright interface is configured.
Step 9	**vrf forwarding** *table-name* **Example:** `Device(config-if)# vrf forwarding VRFRIGHT`	Configures the VRF table.
Step 10	**ip address** {*ip-address mask* [**secondary**] | **pool** *pool-name*} **Example:** `Device(config-if)# ip address 192.168.1.1 255.255.255.0`	Configures a primary or secondary IP address for an interface.
Step 11	**exit** **Example:**	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 12	**ip route** [**vrf** *vrf-name*] *destination-prefix destination-prefix-mask interface-type interface-number* **Example:** `Device(config)# ip route vrf VRFRIGHT 10.0.0.0 255.0.0.0 VASIRIGHT 100`	Establishes a static route for a VRF instance and a VASI interface. **Note** To add an IP route for a VRF instance, you must specify the **vrf** keyword.
Step 13	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for the VRF-Aware Software Infrastructure

Example: Configuring a VASI Interface Pair

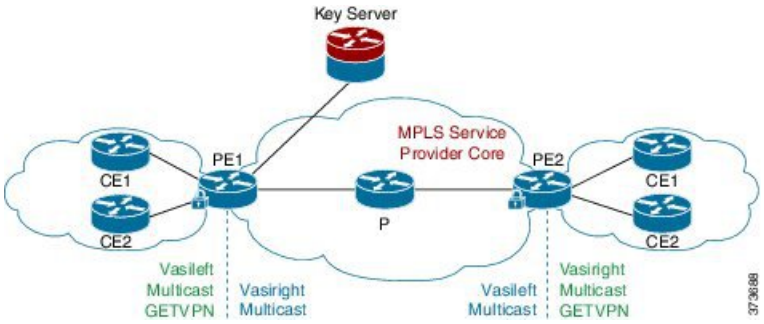
A virtual routing and forwarding (VRF) instance must be enabled for each interface of the VASI pair (VASILEFT and VASIRIGHT). The below example shows how to configure a VASI interface pair.




```
Device(config)# interface vasileft 100
Device(config-if)# vrf forwarding VRFLEFT
Device(config-if)# ip address 192.168.0.1 255.255.255.0
Device(config-if)# exit
Device(config)# ip route vrf VRFLEFT 172.16.0.0 255.255.0.0 vasileft 100
Device(config)# interface vasiright 100
Device(config-if)# vrf forwarding VRFRIGHT
Device(config-if)# ip address 192.168.1.1 255.255.255.0
Device(config-if)# exit
Device(config)# ip route vrf VRFRIGHT 10.0.0.0 255.0.0.0 vasiright 100
Device(config)# end
```

Example: Configuring Multicast and MVPN on VASI

Figure 25: GRE-Based MVPN and GETVPN Configuration



The following example shows how to configure generic routing encapsulation (GRE)-based Multicast VPN (MVPN) and GETVPN on VASI interface pairs. Here, the cryptomap is applied to the vasileft interface. The vasileft interface acts as the customer edge (CE) device and does encryption; the interface is part of the vrf-cust1 virtual routing and forwarding (VRF) instance. The vasiright interface is part of the vrf-core1 VRF instance, to pass traffic across the Multiprotocol Label Switching (MPLS) core and for applied crypto services. The core network supports multicast, and multicast in the VRFs is in stateful switchover (SSO) mode.

```
! PE1 Configuration
Device(config)# vrf definition Mgmt-intf
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
!
Device(config)# vrf definition vrf-core1
Device(config-vrf)# rd 2:1
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# mdt default 203.0.113.1 ! Enables GRE-based MVPN and mdt default
tree
Device(config-vrf-af)# mdt data 203.0.113.33 255.255.255.224 ! Enables the mdt data tree
Device(config-vrf-af)# route-target export 2:1
Device(config-vrf-af)# route-target import 2:1
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# mdt default 203.0.113.1
Device(config-vrf-af)# mdt data 203.0.113.33 255.255.255.224
Device(config-vrf-af)# route-target export 2:1
Device(config-vrf-af)# route-target import 2:1
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
!
Device(config)# vrf definition vrf-cust1
Device(config-vrf)# rd 1:1
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
!
Device(config)# logging buffered 10000000
Device(config)# no logging console
```

```
!
Device(config)# no aaa new-model
Device(config)# clock timezone CST 8 0
!
Device(config)# ip multicast-routing distributed
Device(config)# ip multicast-routing vrf vrf-core1 distributed
Device(config)# ip multicast-routing vrf vrf-cust1 distributed
!
Device(config)# ipv6 unicast-routing
Device(config)# ipv6 multicast-routing
Device(config)# ipv6 multicast-routing vrf vrf-core1
Device(config)# ipv6 multicast-routing vrf vrf-cust1
!
Device(config)# subscriber templating
Device(config)# mpls label protocol ldp
Device(config)# multilink bundle-name authenticated
Device(config)# spanning-tree extend system-id
!
Device(config)# cdp run
Device(config)# ip ftp source-interface GigabitEthernet 0
Device(config)# ip tftp source-interface GigabitEthernet 0
Device(config)# ip tftp blocksize 8192
!
Device(config)# class-map match-any maincampus-ratelimit
Device(config-cmap)# match access-group 101
Device(config-cmap)# exit
!
Device(config)# policy-map transit-limt
Device(config-pmap)# description 160mb transit rate limit
Device(config-pmap)# class maincampus-ratelimit
Device(config-pmap-c)# police 160000000 30000000 60000000 conform-action transmit
exceed-action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)# exit
Device(config-pmap)# exit
!
Device(config)# crypto keyring vrf-cust1 vrf vrf-cust1 ! enables GETVPN
Device(conf-keyring)# pre-shared-key address 0.0.0.0 0.0.0.0 key cisco
Device(conf-keyring)# exit
!
Device(config)# crypto isakmp policy 1
Device(config-isakmp)# encryption 3des
Device(config-isakmp)# authentication pre-share
Device(config-isakmp)# group 2
Device(config-isakmp)# exit
Device(config)# crypto isakmp key cisco address 10.0.3.2
!
Device(config)# crypto gdoi group secure-wan
Device(config-gkm-group)# identity number 12345
Device(config-gkm-group)# server address ipv4 10.0.3.4
Device(config-gkm-group)# exit
!
Device(config)# crypto gdoi group ipv6 ipv6-secure-wan
Device(config-gkm-group)# identity number 123456
Device(config-gkm-group)# server address ipv4 10.0.3.6
Device(config-gkm-group)# exit
!
Device(config)# crypto map getvpn 1 gdoi
Device(config-crypto-map)# set group secure-wan
Device(config-crypto-map)# exit
!
Device(config)# crypto map ipv6 getvpn-v6 1 gdoi
```

```
Device(config-crypto-map)# set group ipv6-secure-wan
Device(config-crypto-map)# exit
!
Device(config)# interface loopback 0
Device(config-if)# ip address 198.51.100.241 255.255.255.240
Device(config-if)# ip pim sparse-mode
Device(config-if)# ipv6 address 2001:DB8::1/32
Device(config-if)# ipv6 enable
Device(config-if)# ospfv3 100 ipv6 area 0
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# vrf forwarding vrf-cust1
Device(config-if)# ip address 192.0.2.1 255.255.255.240
Device(config-if)# shutdown
Device(config-if)# negotiation auto
!
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/0
Device(config-if)# ip address 192.0.2.18 255.255.255.240
Device(config-if)# ip pim sparse-mode
Device(config-if)# negotiation auto
Device(config-if)# mpls ip
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/1
Device(config-if)# vrf forwarding vrf-cust1
Device(config-if)# ip address 10.0.3.1 255.255.255.0
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/2
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/3
Device(config-if)# vrf forwarding vrf-cust1
Device(config-if)# ip address 192.0.2.34 255.255.255.240
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip igmp version 3
Device(config-if)# negotiation auto
Device(config-if)# ipv6 address 2001:DB8:0000:0000:0000:0000:0000:0001/48
Device(config-if)# ospfv3 100 ipv6 area 0
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/4
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0
Device(config-if)# vrf forwarding Mgmt-intf
Device(config-if)# ip address 10.74.30.161 255.255.255.0
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface vasileft 1 ! On the vasileft interface, enable multicast and
```

```
GETVPN.
Device(config-if)# vrf forwarding vrf-cust1
Device(config-if)# ip address 209.165.202.129 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ipv6 address FE80::CEEF:48FF:FEEA:C501 link-local
Device(config-if)# ipv6 address 2001:B000::2/64
Device(config-if)# ipv6 crypto map getvpn-v6
Device(config-if)# ospfv3 100 ipv6 area 0
Device(config-if)# no keepalive
Device(config-if)# crypto map getvpn
Device(config-if)# exit
!
Device(config)# interface vasiright 1 ! On the vasiright interface, only enable multicast.
Device(config-if)# vrf forwarding vrf-core1
Device(config-if)# ip address 209.165.202.130 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ipv6 address 2001:B000::1/64
Device(config-if)# ospfv3 100 ipv6 area 0
Device(config-if)# no keepalive
Device(config-if)# exit
!
Device(config)# router ospfv3 100
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# redistribute bgp 1
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv6 unicast vrf vrf-cust1
Device(config-router-af)# redistribute bgp 1
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv6 unicast vrf vrf-core1
Device(config-router-af)# redistribute bgp 1
Device(config-router-af)# exit-address-family
!
Device(config)# router ospf 1
Device(config-router)# network 1.1.1.1 0.0.0.0 area 0
Device(config-router)# network 192.0.2.0 0.0.0.255 area 0
Device(config-router)# exit
!
Device(config)# router bgp 1 ! Use BGP routing protocol to broadcast vrf-cust1 routing
entry.
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# neighbor 172.16.0.1 remote-as 1
Device(config-router)# neighbor 172.16.0.1 update-source Loopback0
!
Device(config-router)# address-family ipv4
Device(config-router-af)# neighbor 172.16.0.1 activate
Device(config-router-af)# neighbor 172.16.0.1 send-community both
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family vpnv4
Device(config-router-af)# neighbor 172.16.0.1 activate
Device(config-router-af)# neighbor 172.16.0.1 send-community both
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv4 mdt ! For MVPN neighbor setup
Device(config-router-af)# neighbor 172.16.0.1 activate
Device(config-router-af)# neighbor 172.16.0.1 send-community both
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family vpnv6
Device(config-router-af)# neighbor 192.168.0.1 activate
```

```
Device(config-router-af)# neighbor 192.168.0.1 send-community both
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv4 vrf vrf-core1
Device(config-router-af)# bgp router-id 209.165.202.130
Device(config-router-af)# redistribute connected
Device(config-router-af)# neighbor 209.165.202.129 remote-as 65002
Device(config-router-af)# neighbor 209.165.202.129 local-as 65001 no-prepend replace-as
Device(config-router-af)# neighbor 209.165.202.129 activate
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv6 vrf vrf-core1
Device(config-router-af)# redistribute connected
Device(config-router-af)# redistribute ospf 100 include-connected
Device(config-router-af)# bgp router-id 209.165.202.130
Device(config-router-af)# neighbor 2001:B000::2 remote-as 10000
Device(config-router-af)# neighbor 2001:B000::2 local-as 65000 no-prepend replace-as
Device(config-router-af)# neighbor 2001:B000::2 activate
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv4 vrf vrf-cust1
Device(config-router-af)# bgp router-id 209.165.202.129
Device(config-router-af)# redistribute connected
Device(config-router-af)# neighbor 209.165.202.130 remote-as 65001
Device(config-router-af)# neighbor 209.165.202.130 local-as 65002 no-prepend replace-as
Device(config-router-af)# neighbor 209.165.202.130 activate
Device(config-router-af)# exit-address-family
Device(config-router)# exit
!
Device(config-router)# address-family ipv6 vrf vrf-cust1
Device(config-router-af)# redistribute connected
Device(config-router-af)# redistribute ospf 100 include-connected
Device(config-router-af)# bgp router-id 209.165.202.129
Device(config-router-af)# neighbor 2001:B000::1 remote-as 65000
Device(config-router-af)# neighbor 2001:B000::1 local-as 10000 no-prepend replace-as
Device(config-router-af)# neighbor 2001:B000::1 activate
Device(config-router-af)# exit-address-family
!
Device(config)# ip forward-protocol nd
!
Device(config)# no ip http server
Device(config)# no ip http secure-server
Device(config)# ip pim rp-address 1.1.1.1
Device(config)# ip pim vrf vrf-core1 ssm default
Device(config)# ip pim vrf vrf-cust1 ssm default
Device(config)# ip route 192.0.2.0 255.255.255.240 10.11.12.10
Device(config)# ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 10.74.9.1
!
Device(config)# ip access-list standard bidir
Device(config-std-nacl)# exit
!
Device(config)# access-list 101 deny ip 198.51.100.1 255.255.255.240 198.51.100.177
255.255.255.240
Device(config)# ipv6 router eigrp 300
Device(config-rtr)# passive-interface Loopback 0
Device(config-rtr)# redistribute connected
Device(config-rtr)# exit
!
Device(config)# mpls ldp router-id Loopback 0
Device(config)# control-plane
Device(config-cp)# exit
!
```

```
Device(config)# line con 0
Device(config-line)# exec-timeout 0 0
Device(config-line)# privilege level 15
Device(config-line)# logging synchronous
Device(config-line)# stopbits 1
Device(config-line)# exit
Device(config)# line vty 0 4
Device(config-line)# exec-timeout 0 0
Device(config-line)# privilege level 15
Device(config-line)# logging synchronous
Device(config-line)# no login
Device(config-line)# end
```

Verifying Multicast VASI Configuration

Use the following commands to verify the multicast VRF-Aware Software Infrastructure (VASI) configuration:

SUMMARY STEPS

1. **enable**
2. **show ip mroute**
3. **show ip mroute vrf**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 show ip mroute

Displays the contents of the multicast routing (mroute) table.

Example:

```
Device# show ip mroute

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
```

```
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 203.0.113.1), 04:33:39/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet0/0/2, Forward/Sparse-Dense, 04:33:39/stopped
    GigabitEthernet0/0/0, Forward/Sparse-Dense, 04:33:39/stopped
(10.0.0.3, 203.0.113.1), 04:33:36/00:00:36, flags: T
  Incoming interface: GigabitEthernet0/0/2, RPF nbr 10.1.1.3
  Outgoing interface list:
    GigabitEthernet0/0/0, Forward/Sparse-Dense, 04:33:36/stopped
(10.0.0.1, 203.0.113.1), 04:33:39/00:02:44, flags: T
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 10.1.1.0
  Outgoing interface list:
    GigabitEthernet0/0/2, Forward/Sparse-Dense, 04:33:39/stopped
```

Step 3 **show ip mroute vrf**

Filters the output to display only the contents of the multicast routing table that pertains to the Multicast VPN (MVPN) routing and forwarding (MVRF) instance specified for the *vrf-name* argument.

Example:

```
Device# show ip mroute vrf cust1

(10.2.1.1, 203.1.113.4), 00:40:09/00:02:44, flags: sTI
  Incoming interface: vasileft1, RPF nbr 36.1.1.2
  Outgoing interface list:
    GigabitEthernet0/0/1.1, Forward/Sparse-Dense, 00:40:09/00:02:44
PE1#sh ip mroute vrf cust1-core
(10.2.1.1, 203.1.113.4), 04:22:09/00:02:50, flags: sT
  Incoming interface: Tunnel0, RPF nbr 10.0.0.3
  Outgoing interface list:
    vasiright1, Forward/Sparse-Dense, 04:22:09/00:02:50
PE1#sh ip mroute
(*, 203.1.113.4), 21:08:36/stopped, RP 0.0.0.0, flags: DCZ
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet0/0/0, Forward/Sparse-Dense, 04:27:50/stopped
    MVRF cust1-core, Forward/Sparse-Dense, 21:06:53/stopped
(10.0.0.3, 203.1.113.4), 04:26:53/00:01:22, flags: TZ
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 10.1.1.1
  Outgoing interface list:
    MVRF cust1-core, Forward/Sparse-Dense, 04:26:53/stopped
```

Additional References for Configuring the VRF-Aware Software Infrastructure

Related Documents

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

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

Technical Assistance

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

Feature Information for Configuring the VRF-Aware Software Infrastructure

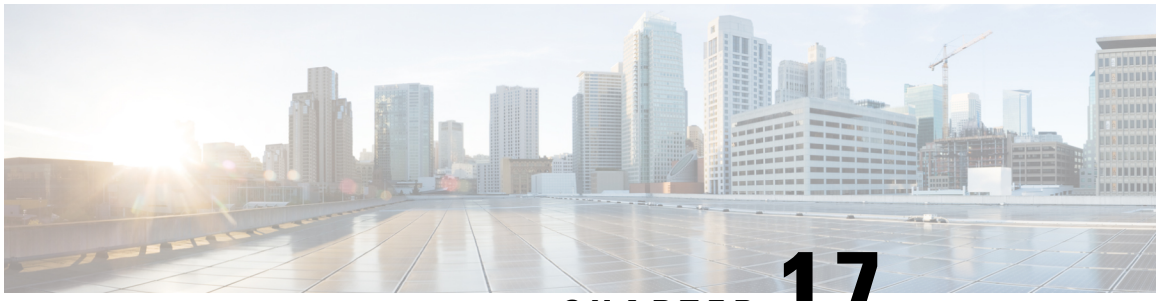
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 19: Feature Information for Configuring the VRF-Aware Software Infrastructure

Feature Name	Releases	Feature Information
Multicast and Multicast VPN on VASI	Cisco IOS XE Release 3.14S	The Multicast and MVPN on VASI feature supports IPv4 and IPv6 multicast and multicast VPN (MVPN) on VASI interfaces. This feature is independent of the multicast modes (sparse, source-specific multicast [SSM] and so on) configured at the customer site and also independent of the MVPN mode—generic routing encapsulation (GRE)-based or Multicast Label Distribution Protocol (MLDP)-based—in the core network. No new commands have been introduced or modified for this feature.

Feature Name	Releases	Feature Information
VRF-Aware Software Infrastructure	Cisco IOS XE Release 2.6	The VRF-Aware Software Infrastructure feature allows you to apply services such as ACLs, NAT, policing, and zone-based firewalls to traffic that flows across two different VRF instances. The VRF-Aware Software Infrastructure (VASI) interfaces support redundancy of the RP and FP. This feature supports IPv4 and IPv6 unicast and multicast traffic on VASI interfaces.
VASI (VRF-Aware Software Infrastructure) Enhancements Phase I	Cisco IOS XE Release 3.1S	The VASI Enhancements Phase I feature provides the following enhancements to VASI: • Support for 500 VASI interfaces. • Support for IBGP dynamic routing between VASI interfaces.
VASI (VRF-Aware Software Infrastructure) Enhancements Phase II	Cisco IOS XE Release 3.2S	The VASI Enhancements Phase II feature provides the following enhancements to VASI: • Support for IPv6 unicast traffic over VASI interfaces. • Support for OSPF and EIGRP dynamic routing between VASI interfaces.
VASI (VRF-Aware Software Infrastructure) Scale	Cisco IOS XE Release 3.3S	The VASI Scale feature provides support for 1000 VASI interfaces. The following command was introduced or modified: **interface (VASI)**.
VASI (VRF-Aware Software Infrastructure) Scale	Cisco IOS XE Release 3.7.2S	The VASI Scale feature provides support for eBGP dynamic routing between VASI interfaces.
VASI 2000 Pair Scale	Cisco IOS XE Release 3.10S	The VASI 2000 Pair Scale feature provides support for 2000 VASI interfaces. 2000 VASI interfaces are supported on Border Gateway Protocol (BGP). The following command was introduced or modified: **interface (VASI)**.



CHAPTER 17

IPv6 Zone-Based Firewall Support over VASI Interfaces

This feature supports VRF-Aware Service Infrastructure (VASI) interfaces over IPv6 firewalls. This feature allows you to apply services such as access control lists (ACLs), Network Address Translation (NAT), policing, and zone-based firewalls to traffic that flows across two different virtual routing and forwarding (VRF) instances. VASI interfaces support the redundancy of Route Processors (RPs) and Forwarding Processors (FPs). VASI interfaces support IPv4 and IPv6 unicast traffic.

This module provides information about VASI interfaces and describes how to configure VASI interfaces.

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 IPv6 Zone-Based Firewall Support over VASI Interfaces

- Multiprotocol Label Switching (MPLS) traffic over VRF-Aware Software Infrastructure (VASI) interfaces is not supported.
- IPv4 and IPv6 multicast traffic is not supported.

- VASI interfaces do not support the attachment of queue-based features. The following commands are not supported on modular QoS CLI (MQC) policies that are attached to VASI interfaces:
  - **bandwidth (policy-map class)**
  - **fair-queue**
  - **priority**
  - **queue-limit**
  - **random-detect**
  - **shape**

Information About IPv6 Zone-Based Firewall Support over VASI Interfaces

VASI Overview

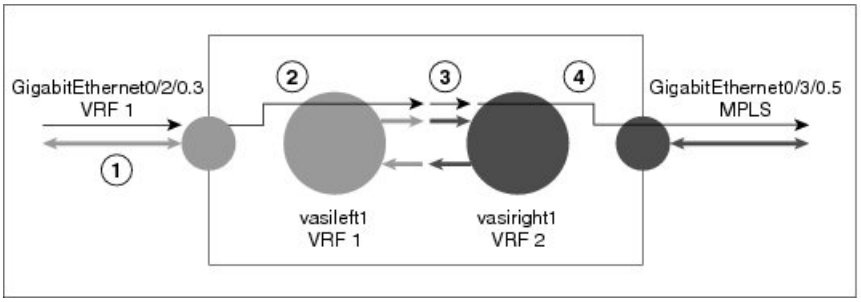
VRF-Aware Software Infrastructure (VASI) provides the ability to apply services such as, a firewall, GETVPN, IPsec, and Network Address Translation (NAT), to traffic that flows across different virtual routing and forwarding (VRF) instances. VASI is implemented by using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF instance. The VASI virtual interface is the next-hop interface for any packet that needs to be switched between these two VRF instances. VASI interfaces provide the framework to configure a firewall or NAT between VRF instances.

Each interface pair is associated with two different VRF instances. The pairing is done automatically based on the two interface indexes such that the vasileft interface is automatically paired to the vasiright interface. For example, in the figure below, vasileft1 and vasiright1 are automatically paired, and a packet entering vasileft1 is internally handed over to vasiright1.

On VASI interfaces, you can configure either static routing or dynamic routing with Internal Border Gateway Protocol (IBGP), Enhanced Interior Gateway Routing Protocol (EIGRP), or Open Shortest Path First (OSPF).

The following figure shows an inter-VRF VASI configuration on the same device.

Figure 26: Inter-VRF VASI Configuration



When an inter-VRF VASI is configured on the same device, the packet flow happens in the following order:

1. A packet enters the physical interface that belongs to VRF 1 (Gigabit Ethernet 0/2/0.3).
2. Before forwarding the packet, a forwarding lookup is done in the VRF 1 routing table. Vasileft1 is chosen as the next hop, and the Time to Live (TTL) value is decremented from the packet. Usually, the forwarding

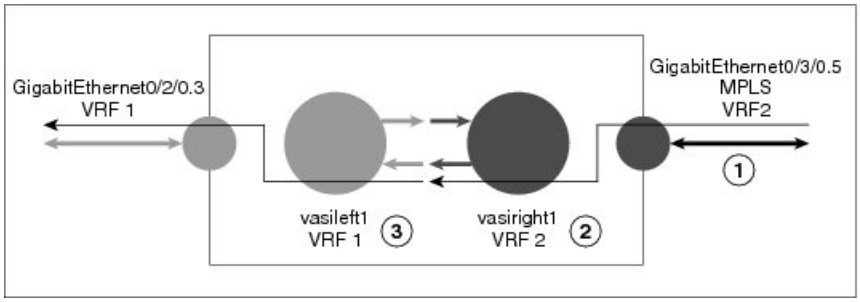
address is selected on the basis of the default route in the VRF. However, the forwarding address can also be a static route or a learned route. The packet is sent to the egress path of vasileft1 and then automatically sent to the vasiright1 ingress path.

3. When the packet enters vasiright1, a forwarding lookup is done in the VRF 2 routing table, and the TTL is decremented again (second time for this packet).
4. VRF 2 forwards the packet to the physical interface, Gigabit Ethernet 0/3/0.5.

The following figure shows how VASI works in a Multiprotocol Label Switching (MPLS) VPN configuration.

Note In the following figure, MPLS is enabled on the Gigabit Ethernet interface, but MPLS traffic is not supported across VASI pairs.

Figure 27: VASI with an MPLS VPN Configuration



When VASI is configured with a Multiprotocol Label Switching (MPLS) VPN, the packet flow happens in the following order:

1. A packet arrives on the MPLS interface with a VPN label.
2. The VPN label is stripped from the packet, a forwarding lookup is done within VRF 2, and the packet is forwarded to vasiright1. The TTL value is decremented from the packet.
3. The packet enters vasileft1 on the ingress path, and another forwarding lookup is done in VRF 1. The packet is sent to the egress physical interface in VRF1 (Gigabit Ethernet 0/2/0.3). The TTL is again decremented from the packet.

How to Configure IPv6 Zone-Based Firewall Support over VASI Interfaces

Configuring VRFs and Address Family Sessions

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **vrf definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
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	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode and enters VRF configuration mode.
Step 6	**end** **Example:** `Device(config-vrf)# end`	Exits VRF configuration mode and enters privileged EXEC mode.

Configuring Class Maps and Policy Maps for VASI Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **class-map type inspect match-any** *class-map-name*
5. **match protocol** *name*
6. **match protocol** *name*
7. **exit**
8. **policy-map type inspect** *policy-map-name*
9. **class type inspect** *class-map-name*

10. **inspect**
11. **exit**
12. **class class-default**
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	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6-unicast routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**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 and enters QoS class-map configuration mode.
Step 5	**match protocol** *name* **Example:** `Device(config-cmap)# match protocol icmp`	Configures a match criterion for a class map on the basis of a specified protocol.
Step 6	**match protocol** *name* **Example:** `Device(config-cmap)# match protocol tcp`	Configures a match criterion for a class map on the basis of a specified protocol.
Step 7	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 8	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p-map`	Creates a protocol-specific inspect-type policy map and enters QoS policy-map configuration mode.
Step 9	**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 QoS policy-map class configuration mode.

	Command or Action	Purpose
Step 10	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 11	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 12	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Applies the policy map settings to the predefined default class and enters QoS policy-map class configuration mode. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 13	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring Zones and Zone Pairs for VASI Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
6. **service-policy type inspect** *policy-map-name*
7. **exit**
8. **interface** *type number*
9. **vrf forwarding** *vrf-name*
10. **no ip address**
11. **zone member security** *zone-name*
12. **ipv6 address** *ipv6-address*/*prefix-length*
13. **ipv6 enable**
14. **negotiation auto**
15. **exit**
16. **interface** *type number*
17. **no ip address**
18. **ipv6 address** *ipv6-address*/*prefix-length*
19. **ipv6 enable**
20. **negotiation auto**
21. **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 in`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source and a destination zone. • In a zone pair, you can use the default zone as either the source or the destination zone.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security in-out source in destination out`	Creates a zone pair and enters security zone-pair configuration mode. • To apply a policy, you must configure a zone pair.
Step 6	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p-map`	Attaches a policy map to a top-level policy map. • If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 7	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 9	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding VRF1`	Associates a virtual routing and forwarding (VRF) instance or a virtual network with an interface or subinterface.

	Command or Action	Purpose
Step 10	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 11	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security in`	Attaches an interface to a security zone.
Step 12	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:2:1234/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 13	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 14	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables advertisement of speed, duplex mode, and flow control on a Gigabit Ethernet interface.
Step 15	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 16	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 17	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 18	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:3:1234/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 19	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 20	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables advertisement of speed, duplex mode, and flow control on a Gigabit Ethernet interface.

	Command or Action	Purpose
Step 21	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring VASI Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ipv6 address** *ipv6-address*/*prefix-length* **link-local**
6. **ipv6 address** *ipv6-address*/*prefix-length*
7. **ipv6 enable**
8. **no keepalive**
9. **zone member security** *zone-name*
10. **exit**
11. **interface** *type number*
12. **ipv6 address** *ipv6-address*/*prefix-length* **link-local**
13. **ipv6 address** *ipv6-address*/*prefix-length*
14. **ipv6 enable**
15. **no keepalive**
16. **exit**
17. **ipv6 route** *ipv6-prefix*/*prefix-length interface-type interface-number ipv6-address*
18. **ipv6 route vrf** *vrf-name ipv6-prefix*/*prefix-length interface-type interface-number ipv6-address*
19. **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 vasileft 1`	Configures a VASI interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding VRF1`	Associates a virtual routing and forwarding (VRF) instance or a virtual network with an interface or subinterface.
Step 5	**ipv6 address** *ipv6-address/prefix-length* **link-local** **Example:** `Device(config-if)# ipv6 address FE80::8EB6:4FFF:FE6C:E701 link-local`	Configures an IPv6 link-local address for an interface and enable IPv6 processing on the interface.
Step 6	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:4:1234/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 7	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 8	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Disables keepalive packets.
Step 9	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security out`	Attaches an interface to a security zone.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface vasiright 1`	Configures a VASI interface and enters interface configuration mode.
Step 12	**ipv6 address** *ipv6-address/prefix-length* **link-local** **Example:** `Device(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local`	Configures an IPv6 link-local address for an interface and enable IPv6 processing on the interface.
Step 13	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:4:1234/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 14	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

	Command or Action	Purpose
Step 15	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Disables keepalive packets.
Step 16	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 17	**ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number ipv6-address* **Example:** `Device(config)# ipv6 route 2001::/64 vasileft 1 2001::/64`	Establishes static IPv6 routes.
Step 18	**ipv6 route vrf** *vrf-name ipv6-prefix/prefix-length interface-type interface-number ipv6-address* **Example:** `Device(config)# ipv6 route vrf vrf1 2001::/64 vasiright 1 2001::/64`	Specifies all VRF tables or a specific VRF table for an IPv6 address.
Step 19	**end** **Example:** `Device(config# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for IPv6 Zone-Based Firewall Support over VASI Interfaces

Example: Configuring VRFs and Address Family Sessions

```
Device# configure terminal
Device(config)# vrf definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# end
```

Example: Configuring Class Maps and Policy Maps for VASI Support

```
Device# configure terminal
Device(config)# ipv6-unicast routing
Device(config)# class-map type inspect match-any c-map
Device(config-cmap)# match protocol icmp
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol udp
```

```
Device(config-cmap)# exit
Device(config)# policy-map type inspect p-map
Device(config-pmap)# class type inspect c-map
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# end
```

Example: Configuring Zones and Zone Pairs for VASI Support

```
Device# configure terminal
Device(config)# zone security in
Device(config)# exit
Device(config)# zone security out
Device(config)# exit
Device(config)# zone-pair security in-out source in destination out
Device(config-sec-zone-pair)# service-policy type inspect p-map
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# vrf forwarding VRF1
Device(config-if)# no ip address
Device(config-if)# zone member security in
Device(config-if)# ipv6 address 2001:DB8:2:1234/64
Device(config-if)# ipv6 enable
Device(config-if)# negotiation auto
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# ipv6 address 2001:DB8:3:1234/64
Device(config-if)# ipv6 enable
Device(config-if)# negotiation auto
Device(config-if)# end
```

Example: Configuring VASI Interfaces

```
Device# configure terminal
Device(config)# interface vasileft 1
Device(config-if)# vrf forwarding VRF1
Device(config-if)# ipv6 address FE80::8EB6:4FFF:FE6C:E701 link-local
Device(config-if)# ipv6 address 2001:DB8:4:1234/64
Device(config-if)# ipv6 enable
Device(config-if)# no keepalive
Device(config-if)# zone-member security out
Device(config-if)# exit
Device(config)# interface vasiright 1
Device(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local
Device(config-if)# ipv6 address 2001:DB8:4:1234/64
Device(config-if)# ipv6 enable
Device(config-if)# no keepalive
Device(config-if)# exit
Device(config)# ipv6 route 2001::/64 vasileft 1 2001::/64
Device(config)# ipv6 route vrf vrf1 2001::/64 vasiright 1 2001::/64
Device(config)# end
```

Additional References for Firewall Stateful Interchassis Redundancy

Related Documents

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

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

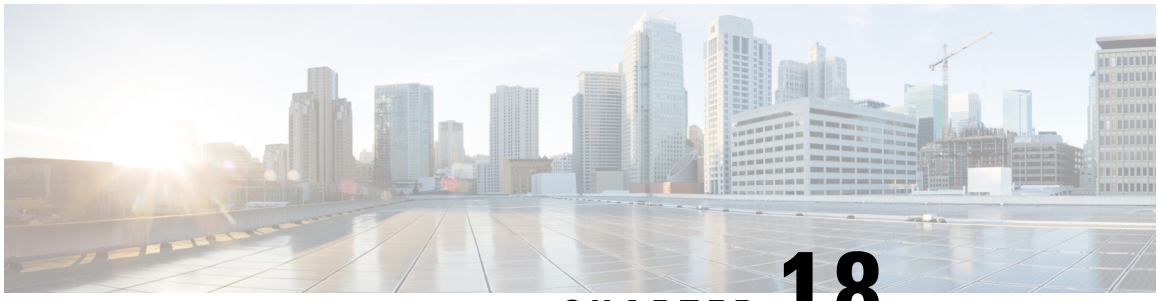
Feature Information for IPv6 Zone-Based Firewall Support over VASI Interfaces

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 20: Feature Information for IPv6 Zone-Based Firewall Support VASI Interfaces

Feature Name	Releases	Feature Information
IPv6 Zone-Based Firewall Support over VASI Interfaces	Cisco IOS XE Release 3.7S	This feature supports VASI interfaces over IPv6 firewalls. This feature allows you to apply services such as access control lists (ACLs), Network Address Translation (NAT), policing, and zone-based firewalls to traffic that flows across two different virtual routing and forwarding (VRF) instances. VASI interfaces support the redundancy of Route Processors (RPs) and Forwarding Processors (FPs). VASI interfaces support IPv4 and IPv6 unicast traffic. No commands were introduced or modified for this feature.



CHAPTER 18

Protection Against Distributed Denial of Service Attacks

The Protection Against Distributed Denial of Service Attacks feature provides protection from Denial of Service (DoS) attacks at the global level (for all firewall sessions) and at the VPN routing and forwarding (VRF) level. In Cisco IOS XE Release 3.4S and later releases, you can configure the aggressive aging of firewall sessions, event rate monitoring of firewall sessions, the half-opened connections limit, and global TCP SYN cookie protection to prevent distributed DoS attacks.

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 Protection Against Distributed Denial of Service Attacks

Aggressive Aging of Firewall Sessions

The Aggressive Aging feature provides the firewall the capability of aggressively aging out sessions to make room for new sessions, thereby protecting the firewall session database from filling. The firewall protects its

resources by removing idle sessions. The Aggressive Aging feature allows firewall sessions to exist for a shorter period of time defined by a timer called aging-out time.

The Aggressive Aging feature includes thresholds to define the start and end of the aggressive aging period—high and low watermarks. The aggressive aging period starts when the session table crosses the high watermark and ends when it falls below the low watermark. During the aggressive aging period, sessions will exist for a shorter period of time that you have configured by using the aging-out time. If an attacker initiates sessions at a rate that is faster than the rate at which the firewall terminates sessions, all resources that are allocated for creating sessions are used and all new connections are rejected. To prevent such attacks, you can configure the Aggressive Aging feature to aggressively age out sessions. This feature is disabled by default.

You can configure aggressive aging for half-opened sessions and total sessions at the box level (box refers to the entire firewall session table) and the virtual routing and forwarding (VRF) level. If you have configured this feature for total sessions, all sessions that consume firewall session resources are taken into account. Total sessions comprise established sessions, half-opened sessions, and sessions in the imprecise session database. (A TCP session that has not yet reached the established state is called a half-opened session.)

A firewall has two session databases: the session database and the imprecise session database. The session database contains sessions with 5-tuple (the source IP address, the destination IP address, the source port, the destination port, and the protocol). A tuple is an ordered list of elements. The imprecise session database contains sessions with fewer than 5-tuple (missing IP addresses, port numbers, and so on). In the case of aggressive aging for half-opened sessions, only half-opened sessions are considered.

You can configure an aggressive aging-out time for Internet Control Message Protocol (ICMP), TCP, and UDP firewall sessions. The aging-out time is set by default to the idle time.

Event Rate Monitoring Feature

The Event Rate Monitoring feature monitors the rate of predefined events in a zone. The Event Rate Monitoring feature includes basic threat detection, which is the ability of a security device to detect possible threats, anomalies, and attacks to resources inside the firewall and to take action against them. You can configure a basic threat detection rate for events. When the incoming rate of a certain type of event exceeds the configured threat detection rate, event rate monitoring considers this event as a threat and takes action to stop the threat. Threat detection inspects events only on the ingress zone (if the Event Rate Monitoring feature is enabled on the ingress zone).

The network administrator is informed about the potential threats via an alert message (syslog or high-speed logger [HSL]) and can take actions such as detecting the attack vector, detecting the zone from which the attack is coming, or configuring devices in the network to block certain behaviors or traffic.

The Event Rate Monitoring feature monitors the following types of events:

- Firewall drops due to basic firewall checks failure—This can include zone or zone-pair check failures, or firewall policies configured with the drop action, and so on.
- Firewall drops due to Layer 4 inspection failure—This can include TCP inspections that have failed because the first TCP packet is not a synchronization (SYN) packet.
- TCP SYN cookie attack—This can include counting the number of SYN packets that are dropped and the number of SYN cookies that are sent as a spoofing attack.

The Event Rate Monitoring feature monitors the average rate and the burst rate of different events. Each event type has a rate object that is controlled by an associated rate that has a configurable parameter set (the average

threshold, the burst threshold, and a time period). The time period is divided into time slots; each time slot is 1/30th of the time period.

The average rate is calculated for every event type. Each rate object holds 30 completed sampling values plus one value to hold the current ongoing sampling period. The current sampling value replaces the oldest calculated value and the average is recalculated. The average rate is calculated during every time period. If the average rate exceeds the average threshold, the Event Rate Monitoring feature will consider this as a possible threat, update the statistics, and inform the network administrator.

The burst rate is implemented by using the token bucket algorithm. For each time slot, the token bucket is filled with tokens. For each event that occurs (of a specific event type), a token is removed from the bucket. An empty bucket means that the burst threshold is reached, and the administrator receives an alarm through the syslog or HSL. You can view the threat detection statistics and learn about possible threats to various events in the zone from the output of the **show policy-firewall stats zone** command.

You must first enable basic threat detection by using the **threat-detection basic-threat** command. Once basic threat detection is configured, you can configure the threat detection rate. To configure the threat detection rate, use the **threat-detection rate** command.

The following table describes the basic threat detection default settings that are applicable if the Event Rate Monitoring feature is enabled.

Table 21: Basic Threat Detection Default Settings

Packet Drop Reason	Threat Detection Settings
Basic firewall drops	average-rate 400 packets per second (pps) burst-rate 1600 pps rate-interval 600 seconds
Inspection-based firewall drops	average-rate 400 pps burst-rate 1600 pps rate-interval 600 seconds
SYN attack firewall drops	average-rate 100 pps burst-rate 200 pps rate-interval 600 seconds

Half-Opened Connections Limit

The firewall session table supports the limiting of half-opened firewall connections. Limiting the number of half-opened sessions will defend the firewall against attacks that might fill the firewall session table at the per-box level or at the virtual routing and forwarding (VRF) level with half-opened sessions and prevent sessions from being established. The half-opened connection limit can be configured for Layer 4 protocols, Internet Control Message Protocol (ICMP), TCP, and UDP. The limit set to the number of UDP half-opened sessions will not affect the TCP or ICMP half-opened sessions. When the configured half-opened session limit is exceeded, all new sessions are rejected and a log message is generated, either in syslog or in the high-speed logger (HSL).

The following sessions are considered as half-opened sessions:

- TCP sessions that have not completed the three-way handshake.
- UDP sessions that have only one packet detected in the UDP flow.
- ICMP sessions that do not receive a reply to the ICMP echo request or the ICMP time-stamp request.

TCP SYN-Flood Attacks

You can configure the global TCP SYN-flood limit to limit SYN flood attacks. TCP SYN-flooding attacks are a type of denial of service (DoS) attack. When the configured TCP SYN-flood limit is reached, the firewall verifies the source of sessions before creating more sessions. Usually, TCP SYN packets are sent to a targeted end host or a range of subnet addresses behind the firewall. These TCP SYN packets have spoofed source IP addresses. A spoofing attack is when a person or program tries to use false data to gain access to resources in a network. TCP SYN flooding can take up all resources on a firewall or an end host, thereby causing denial of service to legitimate traffic. You can configure TCP SYN-flood protection at the VRF level and the zone level.

SYN flood attacks are divided into two types:

- Host flood—SYN flood packets are sent to a single host intending to utilize all resources on that host.
- Firewall session table flood—SYN flood packets are sent to a range of addresses behind the firewall, with the intention of exhausting the session table resources on the firewall, thereby denying resources to the legitimate traffic going through the firewall.

How to Configure Protection Against Distributed Denial of Service Attacks

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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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 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 the Aggressive Aging of Firewall Sessions

You can configure the Aggressive Aging feature for per-box (per-box refers to the entire firewall session table), default-VRF, and per-VRF firewall sessions. Before the Aggressive Aging feature can work, you must configure the aggressive aging and the aging-out time of firewall sessions.

Perform the following tasks to configure the aggressive aging of firewall sessions.

Configuring per-Box Aggressive Aging

Per-box refers to the entire firewall session table. Any configuration that follows the **parameter-map type inspect-global** command applies to the box.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **per-box max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}

5. **per-box aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **end**
10. **show policy-firewall stats 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	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Steps 4 and 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**per-box max-incomplete** *number* **aggressive-aging high** {*value* **low** *value*	**percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# per-box max-incomplete 2000 aggressive-aging high 1500 low 1200`	Configures the maximum limit and the aggressive aging rate for half-opened sessions in the firewall session table.
Step 5	**per-box aggressive-aging high** {*value* **low** *value*	**percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# per-box aggressive-aging high 1700 low 1300`	Configures the aggressive aging limit of total sessions.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • After aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 10	**show policy-firewall stats global** **Example:** `Device# show policy-firewall stats global`	Displays global firewall statistics information.

Configuring Aggressive Aging for a Default VRF

When you configure the **max-incomplete aggressive-aging** `command, it applies to the default VRF.`

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enters one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
5. **session total** *number* [**aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}]
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*

8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **end**
10. **show policy-firewall stats vrf 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	Enters one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# max-incomplete 3455 aggressive-aging high 2345 low 2255`	Configures the maximum limit and the aggressive aging limit of half-opened firewall sessions.
Step 5	**session total** *number* [**aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}] **Example:** `Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60`	Configures the total limit and the aggressive aging limit for total firewall sessions.
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • After aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 10	**show policy-firewall stats vrf global** **Example:** `Device# show policy-firewall stats vrf global`	Displays global VRF firewall policy statistics.

Configuring the Aging Out of Firewall Sessions

You can configure the aging out of ICMP, TCP, or UDP firewall sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
5. **exit**
6. **parameter-map type inspect** *parameter-map-name*
7. **tcp idle-time** *seconds* [**ageout-time** *seconds*]
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **exit**
10. **policy-map type inspect** *policy-map-name*
11. **class type inspect match-any** *class-map-name*
12. **inspect** *parameter-map-name*
13. **end**
14. **show policy-firewall stats vrf** *vrf-pmap-name*

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	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspectglobal`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 4 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF with a parameter map.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 6	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 7	**tcp idle-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp idle-time 3000 ageout-time 100`	Configures the timeout for idle TCP sessions and the aggressive aging-out time for TCP sessions. • You can also configure the **tcp finwait-time** command to specify how long a TCP session will be managed after the firewall detects a finish (FIN) exchange, or you can configure the **tcp synwait-time** command to specify how long the software will wait

	Command or Action	Purpose
		for a TCP session to reach the established state before dropping the session.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • When aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is enabled when the connections drop below the low watermark.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**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 11	**class type inspect match-any** *class-map-name* **Example:** `Device(config-pmap)# class type inspect match-any ddos-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 12	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap1`	Enables stateful packet inspection for the parameter map.
Step 13	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.
Step 14	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Example

The following is sample output from the **show policy-firewall stats vrf vrf1-pmap** command:

```
Device# show policy-firewall stats vrf vrf1-pmap

 VRF: vrf1, Parameter-Map: vrf1-pmap
  Interface reference count: 2
```

```
Total Session Count(estab + half-open): 270, Exceed: 0
Total Session Aggressive Aging Period Off, Event Count: 0

         Half Open
Protocol Session Cnt     Exceed
-------- -----------     ------
All      0               0
UDP      0               0
ICMP     0               0
TCP      0               0

TCP Syn Flood Half Open Count: 0, Exceed: 12
Half Open Aggressive Aging Period Off, Event Count: 0
```

Configuring per-VRF Aggressive Aging

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **exit**
8. **parameter-map type inspect-vrf** *vrf-pmap-name*
9. **max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
10. **session total** *number* [**aggressive-aging** {**high** *value* **low** *value* | **percent** *percent* **low percent** *percent*}]
11. **alert on**
12. **exit**
13. Enter one of the following commands:
    - **parameter-map type inspect-global**
    - **parameter-map type inspect global**
14. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
15. **exit**
16. **parameter-map type inspect** *parameter-map-name*
17. **tcp idle-time** *seconds* [**ageout-time** *seconds*]
18. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
19. **exit**
20. **policy-map type inspect** *policy-map-name*
21. **class type inspect match-any** *class-map-name*
22. **inspect** *parameter-map-name*
23. **end**
24. **show policy-firewall stats vrf** *vrf-pmap-name*

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 vrf** *vrf-name* **Example:** `Device(config)# ip vrf ddos-vrf1`	Defines a VRF instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:2`	Specifies a route distinguisher (RD) for a VRF instance.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:2`	Creates a route-target extended community and exports the routing information to the target VPN extended community.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 100:2`	Creates a route-target extended community and imports routing information from the target VPN extended community.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 8	**parameter-map type inspect-vrf** *vrf-pmap-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect VRF-type parameter map and enters parameter-map type inspect configuration mode.
Step 9	**max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# max-incomplete 2000 aggressive-aging high 1500 low 1200`	Configures the maximum limit and the aggressive aging limit for half-opened sessions.
Step 10	**session total** *number* [**aggressive-aging** {**high** *value* **low** *value* | **percent** *percent* **low percent** *percent*}] **Example:** `Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60`	Configures the total session limit and the aggressive aging limit for the total sessions. • You can configure the total session limit as an absolute value or as a percentage.

	Command or Action	Purpose
Step 11	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 12	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

Step 13	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 14 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 14	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF with a parameter map.
Step 15	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 16	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 17	**tcp idle-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp idle-time 3000 ageout-time 100`	Configures the timeout for idle TCP sessions and the aggressive aging-out time for TCP sessions.
Step 18	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • When aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example,

	Command or Action	Purpose
		instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 19	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 20	**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 21	**class type inspect match-any** *class-map-name* **Example:** `Device(config-pmap)# class type inspect match-any ddos-class`	Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 22	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap1`	Enables stateful packet inspection for the parameter map.
Step 23	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.
Step 24	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Example

The following is sample output from the **show policy-firewall stats vrf vrf1-pmap** command:

```
Device# show policy-firewall stats vrf vrf1-pmap

VRF: vrf1, Parameter-Map: vrf1-pmap
  Interface reference count: 2
       Total Session Count(estab + half-open): 80, Exceed: 0
       Total Session Aggressive Aging Period Off, Event Count: 0

                Half Open
       Protocol Session Cnt     Exceed
       -------- -----------     ------
       All      0               0
       UDP      0               0
       ICMP     0               0
       TCP      0               0
```

```
TCP Syn Flood Half Open Count: 0, Exceed: 116
Half Open Aggressive Aging Period Off, Event Count: 0
```

Configuring Firewall Event Rate Monitoring

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-zone** *zone-pmap-name*
4. **alert on**
5. **threat-detection basic-threat**
6. **threat-detection rate fw-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
7. **threat-detection rate inspect-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
8. **threat-detection rate syn-attack average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
9. **exit**
10. **zone security** *security-zone-name*
11. **protection** *parameter-map-name*
12. **exit**
13. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
14. **end**
15. **show policy-firewall stats zone**

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-zone** *zone-pmap-name* **Example:** `Device(config)# parameter-map type inspect-zone zone-pmap1`	Configures an inspect-zone parameter map and enters parameter-map type inspect configuration mode.

	Command or Action	Purpose
Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages for a zone. • You can use the **log** command to configure the logging of alerts either to the syslog or to the high-speed logger (HSL).
Step 5	**threat-detection basic-threat** **Example:** `Device(config-profile)# threat-detection basic-threat`	Configures basic threat detection for a zone.
Step 6	**threat-detection rate fw-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate fw-drop average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for firewall drop events. • You must configure the **threat-detection basic-threat** command before you configure the **threat-detection rate** command.
Step 7	**threat-detection rate inspect-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate inspect-drop average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for firewall inspection-based drop events.
Step 8	**threat-detection rate syn-attack average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate syn-attack average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for TCP SYN attack events.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**zone security** *security-zone-name* **Example:** `Device(config)# zone security public`	Creates a security zone and enters security zone configuration mode.
Step 11	**protection** *parameter-map-name* **Example:** `Device(config-sec-zone)# protection zone-pmap1`	Attaches the inspect-zone parameter map to the zone and applies the features configured in the inspect-zone parameter map to the zone.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 13	**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 14	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and enters privileged EXEC mode.
Step 15	**show policy-firewall stats zone** **Example:** `Device# show policy-firewall stats zone`	Displays policy firewall statistics at the zone level.

Configuring the per-Box Half-Opened Session Limit

Per-box refers to the entire firewall session table. Any configuration that follows the **parameter-map type inspect-global** command applies to the box.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **alert on**
5. **per-box max-incomplete** *number*
6. **session total** *number*
7. **end**
8. **show policy-firewall stats global**

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	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip to Steps 5 and 6 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**per-box max-incomplete** *number* **Example:** `Device(config-profile)# per-box max-incomplete 12345`	Configures the maximum number of half-opened connections for the firewall session table.
Step 6	**session total** *number* **Example:** `Device(config-profile)# session total 34500`	Configures the total session limit for the firewall session table.
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 8	**show policy-firewall stats global** **Example:** `Device# show policy-firewall stats global`	Displays global firewall statistics information.

Configuring the Half-Opened Session Limit for an Inspect-VRF Parameter Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-name*
4. **alert on**
5. **max-incomplete** *number*
6. **session total** *number*
7. **exit**
8. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
9. **alert on**
10. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
11. **end**
12. **show policy-firewall stats vrf** *vrf-pmap-name*

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-vrf** *vrf-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect-VRF parameter map and enters parameter-map type inspect configuration mode.
Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**max-incomplete** *number* **Example:** `Device(config-profile)# max-incomplete 2000`	Configures the maximum number of half-opened connections per VRF.
Step 6	**session total** *number* **Example:**	Configures the total session limit for a VRF.

	Command or Action	Purpose
	`Device(config-profile)# session total 34500`	
Step 7	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

Step 8	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, you can use either the **parameter-map type inspect-global** command or the **parameter-map type inspect global** command. You cannot configure both these commands together.
		• Skip Step 10 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 9	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 10	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds the VRF to the global parameter map.
Step 11	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 12	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Configuring the Global TCP SYN Flood Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:

- **parameter-map type inspect-global**
- **parameter-map type inspect global**

4. **alert on**
5. **per-box tcp syn-flood limit** *number*
6. **end**
7. **show policy-firewall stats vrf 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	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, you can configure either the **parameter-map type inspect-global** command or the **parameter-map type inspect global** command. You cannot configure both these commands together.
		• Skip Step 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**per-box tcp syn-flood limit** *number* **Example:** `Device(config-profile)# per-box tcp syn-flood limit 500`	Limits the number of TCP half-opened sessions that trigger SYN cookie processing for new SYN packets.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show policy-firewall stats vrf global** **Example:** `Device# show policy-firewall stats vrf global`	(Optional) Displays the status of the global VRF firewall policy. • The command output also displays how many TCP half-opened sessions are present.

Example

The following is sample output from the **show policy-firewall stats vrf global** command:

```
Device# show policy-firewall stats vrf global

Global table statistics
       total_session_cnt: 0
       exceed_cnt:        0
       tcp_half_open_cnt: 0
       syn_exceed_cnt:    0
```

Configuration Examples for Protection Against Distributed Denial of Service Attacks

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 the Aggressive Aging of Firewall Sessions

Example: Configuring per-Box Aggressive Aging

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# per-box max-incomplete 2000 aggressive-aging 1500 low 1200
Device(config-profile)# per-box aggressive-aging high 1700 low 1300
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# end
```

Example: Configuring Aggressive Aging for a Default VRF

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# max-incomplete 2000 aggressive-aging high 1500 low 1200
Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# end
```

Example: Configuring the Aging Out of Firewall Sessions

```
Device# configure terminal
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp idle-time 3000 ageout-time 100
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# exit
Device(config)# policy-map type inspect ddos-fw
Device(config-profile)# class type inspect match-any ddos-class
Device(config-profile)# inspect pmap1
Device(config-profile)# end
```

Example: Configuring per-VRF Aggressive Aging

```
Device# configure terminal
Device(config)# ip vrf ddos-vrf1
Device(config-vrf)# rd 100:2
```

```
Device(config-vrf)# route-target export 100:2
Device(config-vrf)# route-target import 100:2
Device(config-vrf)# exit
Device(config)# parameter-map type inspect-vrf vrf1-pmap
Device(config-profile)# max-incomplete 3455 aggressive-aging high 2345 low 2255
Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60
Device(config-profile)# alert on
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp idle-time 3000 ageout-time 100
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# exit
Device(config)# policy-map type inspect ddos-fw
Device(config-pmap)# class type inspect match-any ddos-class
Device(config-pmap-c)# inspect pmap1
Device(config-profile)# end
```

Example: Configuring Firewall Event Rate Monitoring

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect zone zone-pmap1
Device(config-profile)# alert on
Device(config-profile)# threat-detection basic-threat
Device(config-profile)# threat-detection rate fw-drop average-time-frame 600 average-threshold
 100 burst-threshold 100
Device(config-profile)# threat-detection rate inspect-drop average-time-frame 600
average-threshold 100 burst-threshold 100
Device(config-profile)# threat-detection rate syn-attack average-time-frame 600
average-threshold 100 burst-threshold 100
Device(config-profile)# exit
Device(config)# zone security public
Device(config-sec-zone)# protection zone-pmap1
Device(config-sec-zone)# exit
Device(config)# zone-pair security private2public source private destination public
Device(config-sec-zone-pair)# end
```

Example: Configuring the per-Box Half-Opened Session Limit

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# per-box max-incomplete 12345
Device(config-profile)# session total 34500
Device(config-profile)# end
```

Example: Configuring the Half-Opened Session Limit for an Inspect VRF Parameter Map

```
Device# configure terminal
Device(config)# parameter-map type inspect vrf vrf1-pmap
Device(config-profile)# alert on
Device(config-profile)# max-incomplete 3500
Device(config-profile)# session total 34500
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# end
```

Example: Configuring the Global TCP SYN Flood Limit

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# per-box tcp syn-flood limit 500
Device(config-profile)# end
```

Additional References for Protection Against Distributed Denial of Service Attacks

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	Cisco IOS Security Command Reference
Firewall resource management	*Configuring Firewall Resource Management feature*
Firewall TCP SYN cookie	*Configuring Firewall TCP SYN Cookie 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

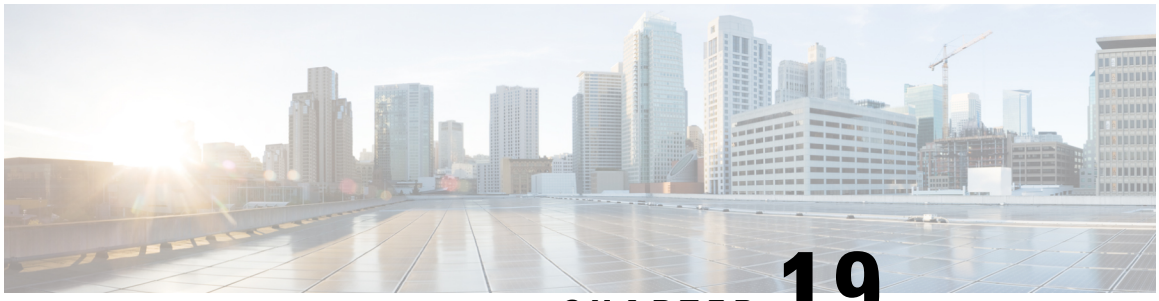
Feature Information for Protection Against Distributed Denial of Service Attacks

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 Protection Against Distributed Denial of Service Attacks

Feature Name	Releases	Feature Information
Protection Against Distributed Denial of Service Attacks	Cisco IOS XE Release 3.4S	The Protection Against Distributed Denial of Service Attacks feature provides protection from DoS attacks at the per-box level (for all firewall sessions) and at the VRF level. You can configure the aggressive aging of firewall sessions, event rate monitoring of firewall sessions, the half-opened connections limit, and global TCP SYN cookie protection to prevent DDoS attacks. The following commands were introduced or modified: **clear policy-firewall stats global**, **max-incomplete**, **max-incomplete aggressive-aging**, **per-box aggressive-aging**, **per-box max-incomplete**, **per-box max-incomplete aggressive-aging**, **per-box tcp syn-flood limit**, **session total**, **show policy-firewall stats global**, **show policy-firewall stats zone**, **threat-detection basic-threat**, **threat-detection rate**, and **udp half-open**.



CHAPTER 19

Configuring Firewall Resource Management

The Firewall Resource Management feature limits the number of VPN Routing and Forwarding (VRF) and global firewall sessions that are configured on a router.

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 Configuring Firewall Resource Management

- After you configure the global-level or VRF-level session limit and reconfigure the session limit, if the global-level or VRF-level session limit is below the initially configured session count, no new session is added; however, no current session is dropped.

Information About Configuring Firewall Resource Management

Firewall Resource Management

Resource Management limits the level of usage of shared resources on a device. Shared resources on a device include:

- Bandwidth
- Connection states
- Memory usage (per table)
- Number of sessions or calls
- Packets per second
- Ternary content addressable memory (TCAM) entries

The Firewall Resource Management feature extends the zone-based firewall resource management from the class level to the VRF level and the global level. Class-level resource management provides resource protection for firewall sessions at a class level. For example, parameters such as the maximum session limit, the session rate limit, and the incomplete session limit protect firewall resources (for example, chunk memory) and keep these resources from being used up by a single class.

When virtual routing and forwarding (VRF) instances share the same policy, a firewall session setup request from one VRF instance can make the total session count reach the maximum limit. When one VRF consumes the maximum amount of resources on a device, it becomes difficult for other VRF instances to share device resources. To limit the number of VRF firewall sessions, you can use the Firewall Resource Management feature.

At the global level, the Firewall Resource Management feature helps limit the usage of resources at the global routing domain by firewall sessions.

VRF-Aware Cisco IOS XE Firewall

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

The VRF-Aware Cisco IOS XE Firewall supports VRF-lite (also known as Multi-VRF CE) and Application Inspection and Control (AIC) for various protocols.

The VRF-aware firewall supports VRF-lite (also known as Multi-VRF CE) and Application Inspection and Control (AIC) for various protocols.

Note Cisco IOS XE Releases do not support Context-Based Access Control (CBAC) firewalls.

Firewall Sessions

Session Definition

At the virtual routing and forwarding (VRF) level, the Firewall Resource Management feature tracks the firewall session count for each VRF instance. At the global level, the firewall resource management tracks the total firewall session count at the global routing domain and not at the device level. In both the VRF and global levels, session count is the sum of opened sessions, half-opened sessions, and sessions in the imprecise firewall session database. A TCP session that has not yet reached the established state is called a half-opened session.

A firewall has two session databases: the session database and the imprecise session database. The session database contains sessions with 5-tuple (source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements. The imprecise session database contains sessions with fewer than 5-tuple (missing IP addresses, port numbers, and so on).

The following rules apply to the configuration of a session limit:

- The class-level session limit can exceed the global limit.
- The class-level session limit can exceed its associated VRF session maximum.
- The sum of the VRF limit, including the global context, can be greater than the hardcoded session limit.

Session Rate

The session rate is the rate at which sessions are established at any given time interval. You can define maximum and minimum session rate limits. When the session rate exceeds the maximum specified rate, the firewall starts rejecting new session setup requests.

From the resource management perspective, setting the maximum and minimum session rate limit helps protect Cisco Packet Processor from being overwhelmed when numerous firewall session setup requests are received.

Incomplete or Half-Opened Sessions

Incomplete sessions are half-opened sessions. Any resource used by an incomplete session is counted, and any growth in the number of incomplete sessions is limited by setting the maximum session limit.

Firewall Resource Management Sessions

The following rules apply to firewall resource management sessions:

- By default, the session limit for opened and half-opened sessions is unlimited.
- Opened or half-opened sessions are limited by parameters and counted separately.
- Opened or half-opened session count includes Internet Control Message Protocol (ICMP), TCP, or UDP sessions.
- You can limit the number and rate of opened sessions.
- You can only limit the number of half-opened sessions.

How to Configure Firewall Resource Management

Configuring Firewall Resource Management

Note A global parameter map takes effect on the global routing domain and not at the router level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-pmap-name*
4. **session total** *number*
5. **tcp syn-flood limit** *number*
6. **exit**
7. **parameter-map type inspect-global**
8. **vrf** *vrf-name* **inspect** *parameter-map-name*
9. **exit**
10. **parameter-map type inspect-vrf vrf-default**
11. **session total** *number*
12. **tcp syn-flood limit** *number*
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 inspect-vrf** *vrf-pmap-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect VRF-type parameter map and enters parameter-map type inspect configuration mode.
Step 4	**session total** *number* **Example:** `Device(config-profile)# session total 1000`	Configures the total number of sessions.

	Command or Action	Purpose
Step 5	**tcp syn-flood limit** *number* **Example:** `Device(config-profile)# tcp syn-flood limit 2000`	Limits the number of TCP half-opened sessions that trigger synchronization (SYN) cookie processing for new SYN packets.
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect-global** **Example:** `Device(config)# parameter-map type inspect-global`	Configures a global parameter map and enters parameter-map type inspect configuration mode.
Step 8	**vrf** *vrf-name* **inspect** *parameter-map-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF to the parameter map.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**parameter-map type inspect-vrf vrf-default** **Example:** `Device(config)# parameter-map type inspect-vrf vrf-default`	Configures a default inspect VRF-type parameter map.
Step 11	**session total** *number* **Example:** `Device(config-profile)# session total 6000`	Configures the total number of sessions. • You can configure the **session total** command for an inspect VRF-type parameter map and for a global parameter map. When you configure the **session total** command for an inspect VRF-type parameter map, the sessions are associated with an inspect VRF-type parameter map. The **session total** command is applied to the global routing domain when it is configured for a global parameter-map.
Step 12	**tcp syn-flood limit** *number* **Example:** `Device(config-profile)# tcp syn-flood limit 7000`	Limits the number of TCP half-opened sessions that trigger SYN cookie processing for new SYN packets.
Step 13	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuration Examples for Firewall Resource Management

Example: Configuring Firewall Resource Management

```
Device# configure terminal
Device(config)# parameter-map type inspect-vrf vrf1-pmap
Device(config-profile)# session total 1000
Device(config-profile)# tcp syn-flood limit 2000
Device(config-profile)# exit
Device(config)# parameter-map type inspect-global
Device(config-profile)# vrf vrf1 inspect pmap1
Device(config-profile)# exit
Device(config)# parameter-map type inspect-vrf vrf-default
Device(config-profile)# session total 6000
Device(config-profile)# tcp syn-flood limit 7000
Device(config-profile)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
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
VRF-aware firewall	“VRF-Aware Cisco IOS XE Firewall” module
Zone-based policy firewall	“ Zone-Based Policy Firewall” module

Technical Assistance

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

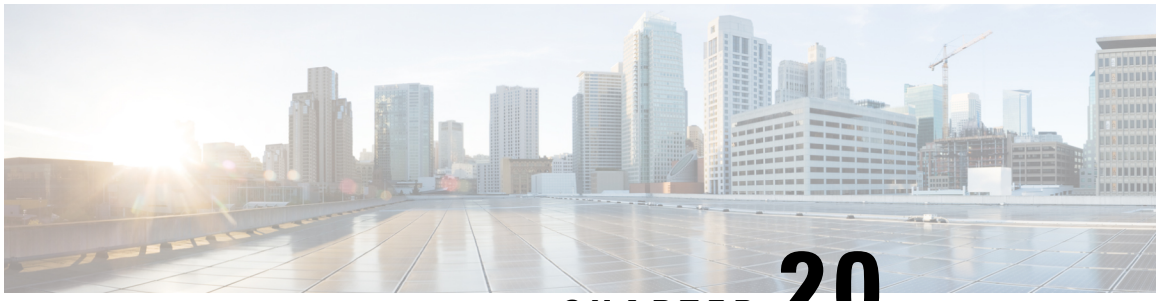
Feature Information for Configuring Firewall Resource Management

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 Configuring Firewall Resource Management

Feature Name	Releases	Feature Information
Firewall Resource Management	Cisco IOS XE Release 3.3S	The Firewall Resource Management feature limits the number of VPN Routing and Forwarding (VRF) and global firewall sessions that are configured on a router. The following commands were introduced or modified: **parameter-map type inspect-vrf**.



CHAPTER 20

IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

IPv6 zone-based firewalls support the Protection of Distributed Denial of Service Attacks and the Firewall Resource Management features.

The Protection Against Distributed Denial of Service Attacks feature provides protection from Denial of Service (DoS) attacks at the global level (for all firewall sessions) and at the VPN routing and forwarding (VRF) level. With the Protection Against Distributed Denial of Service Attacks feature, you can configure the aggressive aging of firewall sessions, event rate monitoring of firewall sessions, half-opened connections limit, and global TCP synchronization (SYN) cookie protection to prevent distributed DoS attacks.

The Firewall Resource Management feature limits the number of VPN Routing and Forwarding (VRF) and global firewall sessions that are configured on a device.

This module describes how to configure the Protection of Distributed Denial of Service Attacks and the Firewall Resource Management features.

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 IPv6 Firewall Support for Protection Against Distributed Denial of Service Attacks and Resource Management

The following restriction applies to the Firewall Resource Management feature:

- After you configure the global-level or the virtual routing and forwarding (VRF)-level session limit and reconfigure the session limit, if the global-level or the VRF-level session limit is below the initially configured session count, no new session is added; however, no current session is dropped.

Information About IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Aggressive Aging of Firewall Sessions

The Aggressive Aging feature provides the firewall the capability of aggressively aging out sessions to make room for new sessions, thereby protecting the firewall session database from filling. The firewall protects its resources by removing idle sessions. The Aggressive Aging feature allows firewall sessions to exist for a shorter period of time defined by a timer called aging-out time.

The Aggressive Aging feature includes thresholds to define the start and end of the aggressive aging period—high and low watermarks. The aggressive aging period starts when the session table crosses the high watermark and ends when it falls below the low watermark. During the aggressive aging period, sessions will exist for a shorter period of time that you have configured by using the aging-out time. If an attacker initiates sessions at a rate that is faster than the rate at which the firewall terminates sessions, all resources that are allocated for creating sessions are used and all new connections are rejected. To prevent such attacks, you can configure the Aggressive Aging feature to aggressively age out sessions. This feature is disabled by default.

You can configure aggressive aging for half-opened sessions and total sessions at the box level (box refers to the entire firewall session table) and the virtual routing and forwarding (VRF) level. If you have configured this feature for total sessions, all sessions that consume firewall session resources are taken into account. Total sessions comprise established sessions, half-opened sessions, and sessions in the imprecise session database. (A TCP session that has not yet reached the established state is called a half-opened session.)

A firewall has two session databases: the session database and the imprecise session database. The session database contains sessions with 5-tuple (the source IP address, the destination IP address, the source port, the destination port, and the protocol). A tuple is an ordered list of elements. The imprecise session database

contains sessions with fewer than 5-tuple (missing IP addresses, port numbers, and so on). In the case of aggressive aging for half-opened sessions, only half-opened sessions are considered.

You can configure an aggressive aging-out time for Internet Control Message Protocol (ICMP), TCP, and UDP firewall sessions. The aging-out time is set by default to the idle time.

Event Rate Monitoring Feature

The Event Rate Monitoring feature monitors the rate of predefined events in a zone. The Event Rate Monitoring feature includes basic threat detection, which is the ability of a security device to detect possible threats, anomalies, and attacks to resources inside the firewall and to take action against them. You can configure a basic threat detection rate for events. When the incoming rate of a certain type of event exceeds the configured threat detection rate, event rate monitoring considers this event as a threat and takes action to stop the threat. Threat detection inspects events only on the ingress zone (if the Event Rate Monitoring feature is enabled on the ingress zone).

The network administrator is informed about the potential threats via an alert message (syslog or high-speed logger [HSL]) and can take actions such as detecting the attack vector, detecting the zone from which the attack is coming, or configuring devices in the network to block certain behaviors or traffic.

The Event Rate Monitoring feature monitors the following types of events:

- Firewall drops due to basic firewall checks failure—This can include zone or zone-pair check failures, or firewall policies configured with the drop action, and so on.
- Firewall drops due to Layer 4 inspection failure—This can include TCP inspections that have failed because the first TCP packet is not a synchronization (SYN) packet.
- TCP SYN cookie attack—This can include counting the number of SYN packets that are dropped and the number of SYN cookies that are sent as a spoofing attack.

The Event Rate Monitoring feature monitors the average rate and the burst rate of different events. Each event type has a rate object that is controlled by an associated rate that has a configurable parameter set (the average threshold, the burst threshold, and a time period). The time period is divided into time slots; each time slot is 1/30th of the time period.

The average rate is calculated for every event type. Each rate object holds 30 completed sampling values plus one value to hold the current ongoing sampling period. The current sampling value replaces the oldest calculated value and the average is recalculated. The average rate is calculated during every time period. If the average rate exceeds the average threshold, the Event Rate Monitoring feature will consider this as a possible threat, update the statistics, and inform the network administrator.

The burst rate is implemented by using the token bucket algorithm. For each time slot, the token bucket is filled with tokens. For each event that occurs (of a specific event type), a token is removed from the bucket. An empty bucket means that the burst threshold is reached, and the administrator receives an alarm through the syslog or HSL. You can view the threat detection statistics and learn about possible threats to various events in the zone from the output of the **show policy-firewall stats zone** command.

You must first enable basic threat detection by using the **threat-detection basic-threat** command. Once basic threat detection is configured, you can configure the threat detection rate. To configure the threat detection rate, use the **threat-detection rate** command.

The following table describes the basic threat detection default settings that are applicable if the Event Rate Monitoring feature is enabled.

Table 24: Basic Threat Detection Default Settings

Packet Drop Reason	Threat Detection Settings
Basic firewall drops	average-rate 400 packets per second (pps) burst-rate 1600 pps rate-interval 600 seconds
Inspection-based firewall drops	average-rate 400 pps burst-rate 1600 pps rate-interval 600 seconds
SYN attack firewall drops	average-rate 100 pps burst-rate 200 pps rate-interval 600 seconds

Half-Opened Connections Limit

The firewall session table supports the limiting of half-opened firewall connections. Limiting the number of half-opened sessions will defend the firewall against attacks that might fill the firewall session table at the per-box level or at the virtual routing and forwarding (VRF) level with half-opened sessions and prevent sessions from being established. The half-opened connection limit can be configured for Layer 4 protocols, Internet Control Message Protocol (ICMP), TCP, and UDP. The limit set to the number of UDP half-opened sessions will not affect the TCP or ICMP half-opened sessions. When the configured half-opened session limit is exceeded, all new sessions are rejected and a log message is generated, either in syslog or in the high-speed logger (HSL).

The following sessions are considered as half-opened sessions:

- TCP sessions that have not completed the three-way handshake.
- UDP sessions that have only one packet detected in the UDP flow.
- ICMP sessions that do not receive a reply to the ICMP echo request or the ICMP time-stamp request.

TCP SYN-Flood Attacks

You can configure the global TCP SYN-flood limit to limit SYN flood attacks. TCP SYN-flooding attacks are a type of denial of service (DoS) attack. When the configured TCP SYN-flood limit is reached, the firewall verifies the source of sessions before creating more sessions. Usually, TCP SYN packets are sent to a targeted end host or a range of subnet addresses behind the firewall. These TCP SYN packets have spoofed source IP addresses. A spoofing attack is when a person or program tries to use false data to gain access to resources in a network. TCP SYN flooding can take up all resources on a firewall or an end host, thereby causing denial of service to legitimate traffic. You can configure TCP SYN-flood protection at the VRF level and the zone level.

SYN flood attacks are divided into two types:

- Host flood—SYN flood packets are sent to a single host intending to utilize all resources on that host.

- Firewall session table flood—SYN flood packets are sent to a range of addresses behind the firewall, with the intention of exhausting the session table resources on the firewall, thereby denying resources to the legitimate traffic going through the firewall.

Firewall Resource Management

Resource Management limits the level of usage of shared resources on a device. Shared resources on a device include:

- Bandwidth
- Connection states
- Memory usage (per table)
- Number of sessions or calls
- Packets per second
- Ternary content addressable memory (TCAM) entries

The Firewall Resource Management feature extends the zone-based firewall resource management from the class level to the VRF level and the global level. Class-level resource management provides resource protection for firewall sessions at a class level. For example, parameters such as the maximum session limit, the session rate limit, and the incomplete session limit protect firewall resources (for example, chunk memory) and keep these resources from being used up by a single class.

When virtual routing and forwarding (VRF) instances share the same policy, a firewall session setup request from one VRF instance can make the total session count reach the maximum limit. When one VRF consumes the maximum amount of resources on a device, it becomes difficult for other VRF instances to share device resources. To limit the number of VRF firewall sessions, you can use the Firewall Resource Management feature.

At the global level, the Firewall Resource Management feature helps limit the usage of resources at the global routing domain by firewall sessions.

Firewall Sessions

Session Definition

At the virtual routing and forwarding (VRF) level, the Firewall Resource Management feature tracks the firewall session count for each VRF instance. At the global level, the firewall resource management tracks the total firewall session count at the global routing domain and not at the device level. In both the VRF and global levels, session count is the sum of opened sessions, half-opened sessions, and sessions in the imprecise firewall session database. A TCP session that has not yet reached the established state is called a half-opened session.

A firewall has two session databases: the session database and the imprecise session database. The session database contains sessions with 5-tuple (source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements. The imprecise session database contains sessions with fewer than 5-tuple (missing IP addresses, port numbers, and so on).

The following rules apply to the configuration of a session limit:

- The class-level session limit can exceed the global limit.
- The class-level session limit can exceed its associated VRF session maximum.
- The sum of the VRF limit, including the global context, can be greater than the hardcoded session limit.

Session Rate

The session rate is the rate at which sessions are established at any given time interval. You can define maximum and minimum session rate limits. When the session rate exceeds the maximum specified rate, the firewall starts rejecting new session setup requests.

From the resource management perspective, setting the maximum and minimum session rate limit helps protect Cisco Packet Processor from being overwhelmed when numerous firewall session setup requests are received.

Incomplete or Half-Opened Sessions

Incomplete sessions are half-opened sessions. Any resource used by an incomplete session is counted, and any growth in the number of incomplete sessions is limited by setting the maximum session limit.

Firewall Resource Management Sessions

The following rules apply to firewall resource management sessions:

- By default, the session limit for opened and half-opened sessions is unlimited.
- Opened or half-opened sessions are limited by parameters and counted separately.
- Opened or half-opened session count includes Internet Control Message Protocol (ICMP), TCP, or UDP sessions.
- You can limit the number and rate of opened sessions.
- You can only limit the number of half-opened sessions.

How to Configure IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Configuring an IPv6 Firewall

The steps to configure an IPv4 firewall and an IPv6 firewall are the same. To configure an IPv6 firewall, you must configure the class map in such a way that only an IPv6 address family is matched.

The **match protocol** command applies to both IPv4 and IPv6 traffic and can be included in either an IPv4 policy or an IPv6 policy.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **vrf-definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **sessions maximum** *sessions*
9. **exit**
10. **ipv6 unicast-routing**
11. **ip port-map** *appl-name* **port** *port-num* **list** *list-name*
12. **ipv6 access-list** *access-list-name*
13. **permit ipv6 any any**
14. **exit**
15. **class-map type inspect match-all** *class-map-name*
16. **match access-group name** *access-group-name*
17. **match protocol** *protocol-name*
18. **exit**
19. **policy-map type inspect** *policy-map-name*
20. **class type inspect** *class-map-name*
21. **inspect** [*parameter-map-name*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf-definition** *vrf-name* **Example:** `Device(config)# vrf-definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect ipv6-param-map`	Enables a global inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
Step 8	**sessions maximum** *sessions* **Example:** `Device(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 11	**ip port-map** *appl-name* **port** *port-num* **list** *list-name* **Example:** `Device(config)# ip port-map ftp port 8090 list ipv6-acl`	Establishes a port to application mapping (PAM) by using the IPv6 access control list (ACL).
Step 12	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 13	**permit ipv6 any any** **Example:** `Device(config-ipv6-acl)# permit ipv6 any any`	Sets permit conditions for an IPv6 access list.
Step 14	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 15	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ipv6-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.
Step 16	**match access-group name** *access-group-name* **Example:**	Configures the match criteria for a class map on the basis of the specified ACL.

	Command or Action	Purpose
	`Device(config-cmap)# match access-group name ipv6-acl`	
Step 17	**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 18	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 19	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ipv6-policy`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 20	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ipv6-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 21	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect ipv6-param-map`	Enables stateful packet inspection.
Step 22	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring the Aggressive Aging of Firewall Sessions

You can configure the Aggressive Aging feature for per-box (per-box refers to the entire firewall session table), default-VRF, and per-VRF firewall sessions. Before the Aggressive Aging feature can work, you must configure the aggressive aging and the aging-out time of firewall sessions.

Perform the following tasks to configure the aggressive aging of firewall sessions.

Configuring per-Box Aggressive Aging

Per-box refers to the entire firewall session table. Any configuration that follows the **parameter-map type inspect-global** command applies to the box.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**

- **parameter-map type inspect global**

4. **per-box max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
5. **per-box aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **end**
10. **show policy-firewall stats 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	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Steps 4 and 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**per-box max-incomplete** *number* **aggressive-aging high** {*value* **low** *value*	**percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# per-box max-incomplete 2000 aggressive-aging high 1500 low 1200`	Configures the maximum limit and the aggressive aging rate for half-opened sessions in the firewall session table.
Step 5	**per-box aggressive-aging high** {*value* **low** *value*	**percent** *percent* **low percent** *percent*}	Configures the aggressive aging limit of total sessions.

	Command or Action	Purpose
	Example: `Device(config-profile)# per-box aggressive-aging high 1700 low 1300`	
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • After aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 10	**show policy-firewall stats global** **Example:** `Device# show policy-firewall stats global`	Displays global firewall statistics information.

Configuring Aggressive Aging for a Default VRF

When you configure the **max-incomplete aggressive-aging** `command, it applies to the default VRF.`

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enters one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}

5. **session total** *number* [**aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}]
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **end**
10. **show policy-firewall stats vrf 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	Enters one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# max-incomplete 3455 aggressive-aging high 2345 low 2255`	Configures the maximum limit and the aggressive aging limit of half-opened firewall sessions.
Step 5	**session total** *number* [**aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}] **Example:** `Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60`	Configures the total limit and the aggressive aging limit for total firewall sessions.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • After aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 10	**show policy-firewall stats vrf global** **Example:** `Device# show policy-firewall stats vrf global`	Displays global VRF firewall policy statistics.

Configuring per-VRF Aggressive Aging

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **exit**
8. **parameter-map type inspect-vrf** *vrf-pmap-name*
9. **max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
10. **session total** *number* [**aggressive-aging** {**high** *value* **low** *value* | **percent** *percent* **low percent** *percent*}]
11. **alert on**
12. **exit**

13. Enter one of the following commands:
    - **parameter-map type inspect-global**
    - **parameter-map type inspect global**
14. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
15. **exit**
16. **parameter-map type inspect** *parameter-map-name*
17. **tcp idle-time** *seconds* [**ageout-time** *seconds*]
18. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
19. **exit**
20. **policy-map type inspect** *policy-map-name*
21. **class type inspect match-any** *class-map-name*
22. **inspect** *parameter-map-name*
23. **end**
24. **show policy-firewall stats vrf** *vrf-pmap-name*

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 vrf** *vrf-name* **Example:** `Device(config)# ip vrf ddos-vrf1`	Defines a VRF instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:2`	Specifies a route distinguisher (RD) for a VRF instance.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:2`	Creates a route-target extended community and exports the routing information to the target VPN extended community.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 100:2`	Creates a route-target extended community and imports routing information from the target VPN extended community.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 8	**parameter-map type inspect-vrf** *vrf-pmap-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect VRF-type parameter map and enters parameter-map type inspect configuration mode.
Step 9	**max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# max-incomplete 2000 aggressive-aging high 1500 low 1200`	Configures the maximum limit and the aggressive aging limit for half-opened sessions.
Step 10	**session total** *number* [**aggressive-aging** {**high** *value* **low** *value* | **percent** *percent* **low percent** *percent*}] **Example:** `Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60`	Configures the total session limit and the aggressive aging limit for the total sessions. • You can configure the total session limit as an absolute value or as a percentage.
Step 11	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 12	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

Step 13	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 14 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 14	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF with a parameter map.

	Command or Action	Purpose
Step 15	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 16	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 17	**tcp idle-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp idle-time 3000 ageout-time 100`	Configures the timeout for idle TCP sessions and the aggressive aging-out time for TCP sessions.
Step 18	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • When aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 19	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 20	**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 21	**class type inspect match-any** *class-map-name* **Example:** `Device(config-pmap)# class type inspect match-any ddos-class`	Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 22	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap1`	Enables stateful packet inspection for the parameter map.
Step 23	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 24	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Example

The following is sample output from the **show policy-firewall stats vrf vrf1-pmap** command:

```
Device# show policy-firewall stats vrf vrf1-pmap

VRF: vrf1, Parameter-Map: vrf1-pmap
  Interface reference count: 2
       Total Session Count(estab + half-open): 80, Exceed: 0
       Total Session Aggressive Aging Period Off, Event Count: 0

                Half Open
       Protocol Session Cnt     Exceed
       -------- -----------     ------
       All      0               0
       UDP      0               0
       ICMP     0               0
       TCP      0               0

       TCP Syn Flood Half Open Count: 0, Exceed: 116
       Half Open Aggressive Aging Period Off, Event Count: 0
```

Configuring the Aging Out of Firewall Sessions

You can configure the aging out of ICMP, TCP, or UDP firewall sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
5. **exit**
6. **parameter-map type inspect** *parameter-map-name*
7. **tcp idle-time** *seconds* [**ageout-time** *seconds*]
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **exit**
10. **policy-map type inspect** *policy-map-name*
11. **class type inspect match-any** *class-map-name*
12. **inspect** *parameter-map-name*
13. **end**
14. **show policy-firewall stats vrf** *vrf-pmap-name*

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	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspectglobal`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 4 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF with a parameter map.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 6	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 7	**tcp idle-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp idle-time 3000 ageout-time 100`	Configures the timeout for idle TCP sessions and the aggressive aging-out time for TCP sessions. • You can also configure the **tcp finwait-time** command to specify how long a TCP session will be managed after the firewall detects a finish (FIN) exchange, or you can configure the **tcp synwait-time** command to specify how long the software will wait

	Command or Action	Purpose
		for a TCP session to reach the established state before dropping the session.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • When aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is enabled when the connections drop below the low watermark.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**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 11	**class type inspect match-any** *class-map-name* **Example:** `Device(config-pmap)# class type inspect match-any ddos-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 12	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap1`	Enables stateful packet inspection for the parameter map.
Step 13	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.
Step 14	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Example

The following is sample output from the **show policy-firewall stats vrf vrf1-pmap** command:

```
Device# show policy-firewall stats vrf vrf1-pmap

 VRF: vrf1, Parameter-Map: vrf1-pmap
  Interface reference count: 2
```

```
Total Session Count(estab + half-open): 270, Exceed: 0
Total Session Aggressive Aging Period Off, Event Count: 0

         Half Open
Protocol Session Cnt     Exceed
-------- -----------     ------
All      0               0
UDP      0               0
ICMP     0               0
TCP      0               0

TCP Syn Flood Half Open Count: 0, Exceed: 12
Half Open Aggressive Aging Period Off, Event Count: 0
```

Configuring Firewall Event Rate Monitoring

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-zone** *zone-pmap-name*
4. **alert on**
5. **threat-detection basic-threat**
6. **threat-detection rate fw-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
7. **threat-detection rate inspect-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
8. **threat-detection rate syn-attack average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
9. **exit**
10. **zone security** *security-zone-name*
11. **protection** *parameter-map-name*
12. **exit**
13. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
14. **end**
15. **show policy-firewall stats zone**

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 inspect-zone** *zone-pmap-name* **Example:** `Device(config)# parameter-map type inspect-zone zone-pmap1`	Configures an inspect-zone parameter map and enters parameter-map type inspect configuration mode.
Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages for a zone. • You can use the **log** command to configure the logging of alerts either to the syslog or to the high-speed logger (HSL).
Step 5	**threat-detection basic-threat** **Example:** `Device(config-profile)# threat-detection basic-threat`	Configures basic threat detection for a zone.
Step 6	**threat-detection rate fw-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate fw-drop average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for firewall drop events. • You must configure the **threat-detection basic-threat** command before you configure the **threat-detection rate** command.
Step 7	**threat-detection rate inspect-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate inspect-drop average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for firewall inspection-based drop events.
Step 8	**threat-detection rate syn-attack average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate syn-attack average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for TCP SYN attack events.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**zone security** *security-zone-name* **Example:** `Device(config)# zone security public`	Creates a security zone and enters security zone configuration mode.

	Command or Action	Purpose
Step 11	**protection** *parameter-map-name* **Example:** `Device(config-sec-zone)# protection zone-pmap1`	Attaches the inspect-zone parameter map to the zone and applies the features configured in the inspect-zone parameter map to the zone.
Step 12	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 13	**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 14	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and enters privileged EXEC mode.
Step 15	**show policy-firewall stats zone** **Example:** `Device# show policy-firewall stats zone`	Displays policy firewall statistics at the zone level.

Configuring the per-Box Half-Opened Session Limit

Per-box refers to the entire firewall session table. Any configuration that follows the **parameter-map type inspect-global** command applies to the box.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **alert on**
5. **per-box max-incomplete** *number*
6. **session total** *number*
7. **end**
8. **show policy-firewall stats global**

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	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip to Steps 5 and 6 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**per-box max-incomplete** *number* **Example:** `Device(config-profile)# per-box max-incomplete 12345`	Configures the maximum number of half-opened connections for the firewall session table.
Step 6	**session total** *number* **Example:** `Device(config-profile)# session total 34500`	Configures the total session limit for the firewall session table.
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 8	**show policy-firewall stats global** **Example:** `Device# show policy-firewall stats global`	Displays global firewall statistics information.

Configuring the Half-Opened Session Limit for an Inspect-VRF Parameter Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-name*
4. **alert on**
5. **max-incomplete** *number*
6. **session total** *number*
7. **exit**
8. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
9. **alert on**
10. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
11. **end**
12. **show policy-firewall stats vrf** *vrf-pmap-name*

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-vrf** *vrf-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect-VRF parameter map and enters parameter-map type inspect configuration mode.
Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**max-incomplete** *number* **Example:** `Device(config-profile)# max-incomplete 2000`	Configures the maximum number of half-opened connections per VRF.
Step 6	**session total** *number* **Example:**	Configures the total session limit for a VRF.

	Command or Action	Purpose
	`Device(config-profile)# session total 34500`	
Step 7	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

Step 8	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, you can use either the **parameter-map type inspect-global** command or the **parameter-map type inspect global** command. You cannot configure both these commands together.
		• Skip Step 10 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 9	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 10	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds the VRF to the global parameter map.
Step 11	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 12	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Configuring the Global TCP SYN Flood Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:

- **parameter-map type inspect-global**
- **parameter-map type inspect global**

4. **alert on**
5. **per-box tcp syn-flood limit** *number*
6. **end**
7. **show policy-firewall stats vrf 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	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, you can configure either the **parameter-map type inspect-global** command or the **parameter-map type inspect global** command. You cannot configure both these commands together.
		• Skip Step 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**per-box tcp syn-flood limit** *number* **Example:** `Device(config-profile)# per-box tcp syn-flood limit 500`	Limits the number of TCP half-opened sessions that trigger SYN cookie processing for new SYN packets.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show policy-firewall stats vrf global** **Example:** `Device# show policy-firewall stats vrf global`	(Optional) Displays the status of the global VRF firewall policy. • The command output also displays how many TCP half-opened sessions are present.

Example

The following is sample output from the **show policy-firewall stats vrf global** command:

```
Device# show policy-firewall stats vrf global

Global table statistics
       total_session_cnt: 0
       exceed_cnt:        0
       tcp_half_open_cnt: 0
       syn_exceed_cnt:    0
```

Configuring Firewall Resource Management

Note A global parameter map takes effect on the global routing domain and not at the router level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-pmap-name*
4. **session total** *number*
5. **tcp syn-flood limit** *number*
6. **exit**
7. **parameter-map type inspect-global**
8. **vrf** *vrf-name* **inspect** *parameter-map-name*
9. **exit**
10. **parameter-map type inspect-vrf vrf-default**
11. **session total** *number*
12. **tcp syn-flood limit** *number*
13. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-vrf** *vrf-pmap-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect VRF-type parameter map and enters parameter-map type inspect configuration mode.
Step 4	**session total** *number* **Example:** `Device(config-profile)# session total 1000`	Configures the total number of sessions.
Step 5	**tcp syn-flood limit** *number* **Example:** `Device(config-profile)# tcp syn-flood limit 2000`	Limits the number of TCP half-opened sessions that trigger synchronization (SYN) cookie processing for new SYN packets.
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect-global** **Example:** `Device(config)# parameter-map type inspect-global`	Configures a global parameter map and enters parameter-map type inspect configuration mode.
Step 8	**vrf** *vrf-name* **inspect** *parameter-map-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF to the parameter map.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**parameter-map type inspect-vrf vrf-default** **Example:** `Device(config)# parameter-map type inspect-vrf vrf-default`	Configures a default inspect VRF-type parameter map.
Step 11	**session total** *number* **Example:** `Device(config-profile)# session total 6000`	Configures the total number of sessions. • You can configure the **session total** command for an inspect VRF-type parameter map and for a global parameter map. When you configure the **session total** command for an inspect VRF-type parameter map, the sessions are associated with an inspect VRF-type

	Command or Action	Purpose
		parameter map. The **session total** command is applied to the global routing domain when it is configured for a global parameter-map.
Step 12	**tcp syn-flood limit** *number* **Example:** `Device(config-profile)# tcp syn-flood limit 7000`	Limits the number of TCP half-opened sessions that trigger SYN cookie processing for new SYN packets.
Step 13	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuration Examples for IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Example: Configuring an IPv6 Firewall

```
Device# configure terminal
Device(config)# vrf-definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# parameter-map type inspect ipv6-param-map
Device(config-profile)# sessions maximum 10000
Device(config-profile)# exit
Device(config)# ipv6 unicast-routing
Device(config)# ip port-map ftp port 8090 list ipv6-acl
Device(config)# ipv6 access-list ipv6-acl
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# class-map type inspect match-all ipv6-class
Device(config-cmap)# match access-group name ipv6-acl
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect ipv6-policy
Device(config-pmap)# class type inspect ipv6-class
Device(config-pmap-c)# inspect ipv6-param-map
Device(config-pmap-c)# end
```

Example: Configuring the Aggressive Aging of Firewall Sessions

Example: Configuring per-Box Aggressive Aging

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# per-box max-incomplete 2000 aggressive-aging 1500 low 1200
Device(config-profile)# per-box aggressive-aging high 1700 low 1300
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# end
```

Example: Configuring Aggressive Aging for a Default VRF

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# max-incomplete 2000 aggressive-aging high 1500 low 1200
Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# end
```

Example: Configuring per-VRF Aggressive Aging

```
Device# configure terminal
Device(config)# ip vrf ddos-vrf1
Device(config-vrf)# rd 100:2
Device(config-vrf)# route-target export 100:2
Device(config-vrf)# route-target import 100:2
Device(config-vrf)# exit
Device(config)# parameter-map type inspect-vrf vrf1-pmap
Device(config-profile)# max-incomplete 3455 aggressive-aging high 2345 low 2255
Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60
Device(config-profile)# alert on
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp idle-time 3000 ageout-time 100
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# exit
Device(config)# policy-map type inspect ddos-fw
Device(config-pmap)# class type inspect match-any ddos-class
Device(config-pmap-c)# inspect pmap1
Device(config-profile)# end
```

Example: Configuring the Aging Out of Firewall Sessions

```
Device# configure terminal
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
```

```
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp idle-time 3000 ageout-time 100
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# exit
Device(config)# policy-map type inspect ddos-fw
Device(config-profile)# class type inspect match-any ddos-class
Device(config-profile)# inspect pmap1
Device(config-profile)# end
```

Example: Configuring Firewall Event Rate Monitoring

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect zone zone-pmap1
Device(config-profile)# alert on
Device(config-profile)# threat-detection basic-threat
Device(config-profile)# threat-detection rate fw-drop average-time-frame 600 average-threshold
 100 burst-threshold 100
Device(config-profile)# threat-detection rate inspect-drop average-time-frame 600
average-threshold 100 burst-threshold 100
Device(config-profile)# threat-detection rate syn-attack average-time-frame 600
average-threshold 100 burst-threshold 100
Device(config-profile)# exit
Device(config)# zone security public
Device(config-sec-zone)# protection zone-pmap1
Device(config-sec-zone)# exit
Device(config)# zone-pair security private2public source private destination public
Device(config-sec-zone-pair)# end
```

Example: Configuring the per-Box Half-Opened Session Limit

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# per-box max-incomplete 12345
Device(config-profile)# session total 34500
Device(config-profile)# end
```

Example: Configuring the Half-Opened Session Limit for an Inspect VRF Parameter Map

```
Device# configure terminal
Device(config)# parameter-map type inspect vrf vrf1-pmap
Device(config-profile)# alert on
Device(config-profile)# max-incomplete 3500
Device(config-profile)# session total 34500
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
```

```
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# end
```

Example: Configuring the Global TCP SYN Flood Limit

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# per-box tcp syn-flood limit 500
Device(config-profile)# end
```

Example: Configuring Firewall Resource Management

```
Device# configure terminal
Device(config)# parameter-map type inspect-vrf vrf1-pmap
Device(config-profile)# session total 1000
Device(config-profile)# tcp syn-flood limit 2000
Device(config-profile)# exit
Device(config)# parameter-map type inspect-global
Device(config-profile)# vrf vrf1 inspect pmap1
Device(config-profile)# exit
Device(config)# parameter-map type inspect-vrf vrf-default
Device(config-profile)# session total 6000
Device(config-profile)# tcp syn-flood limit 7000
Device(config-profile)# end
```

Additional References for IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Related Documents

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

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

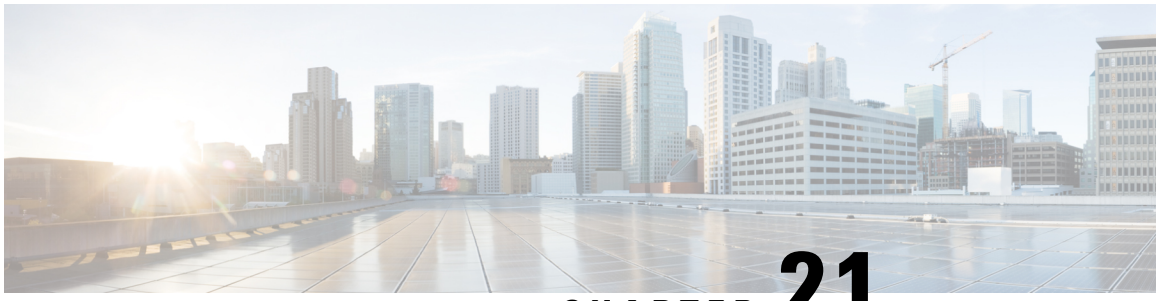
Feature Information for IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

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 25: Feature Information for IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Feature Name	Releases	Feature Information
IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management	Cisco IOS XE Release 3.7S	IPv6 zone-based firewalls support the Protection of Distributed Denial of Service Attacks and the Firewall Resource Management features. The Protection Against Distributed Denial of Service Attacks feature provides protection from Denial of Service (DoS) attacks at the global level (for all firewall sessions) and at the VPN routing and forwarding (VRF) level. You can configure the aggressive aging of firewall sessions, event rate monitoring of firewall sessions, half-opened connections limit, and global TCP SYN cookie protection to prevent distributed DoS attacks. The Firewall Resource Management feature limits the number of VPN routing and forwarding (VRF) instances and global firewall sessions that are configured on a device.
IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management	Cisco IOS XE Release 3.10S	In Cisco IOS XE Release 3.10S, support was added for Cisco CSR 1000V Series Routers.



CHAPTER 21

Configurable Number of Simultaneous Packets per Flow

In zone-based policy firewalls, the number of simultaneous packets per flow is restricted to 25 and packets that exceed the limit are dropped. The dropping of packets when the limit is reached impacts the performance of networks. The Configurable Number of Simultaneous Packets per Flow feature allows you to configure the number of simultaneous packets per flow from 25 to 100.

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

Finding Feature Information

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

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

Restrictions for Configurable Number of Simultaneous Packets per Flow

- When the TCP window scale option is configured, the firewall cannot simultaneously fit too many TCP packets per flow, and packets that exceed the configured limit are dropped. The maximum window size that can be used, if the TCP window scale option is enabled, is 1 GB.

The standard TCP window size is between 2 and 65,535 bytes. If the TCP payload size is smaller than 655 bytes, 100 simultaneous packets cannot contain all TCP packets that belong to a single TCP window, and this can result in packet drops. We recommend that you increase the TCP payload size or reduce the TCP window size to avoid packet drops.

- The total available threads in each platform varies according to the enabled license levels. If the configured number of simultaneous packets per flow is bigger than the available hardware thread number, the configuration of simultaneous packets is not effective.

Information About Configurable Number of Simultaneous Packets per Flow

Overview of Configurable Number of Simultaneous Packets per Flow

The Configurable Number of Simultaneous Packets per Flow feature allows you to increase the number of simultaneous packets per flow that can enter a network. You can increase the number of simultaneous packets per flow from 25 to 100. The default is 25 simultaneous packets.

In multithreaded environments, the zone-based policy firewall may simultaneously receive multiple packets for a single traffic flow. During packet processing, the firewall uses two types of locks: flow lock and software lock. The flow lock ensures that packets that belong to the same flow are processed in the correct order. Normal software locks are used when multiple power processing element (PPE) threads try to read or write critical sections or common data structure (for example, memory).

If the number of simultaneous packets per flow is too large, the time taken by a thread to request and acquire a lock may be too long. This latency adversely affects time-critical infrastructure such as resource reuse and heat-beat processing. To control latency, the number of simultaneous packets was restricted to 25, and packets that exceeded 25 were dropped.

However, the dropping of packets drastically impacts system performance of a system. To minimize packet dropping, the Configurable Number of Simultaneous Packets per Flow feature was introduced. You can configure the number of simultaneous packets per flow from 25 to 100.

To change the number of simultaneous packets per flow, you must configure either the **parameter-map type inspect** *parameter-map-name* command or the **parameter-map type inspect global** command, followed by the **session packet** command. The limit configured under the **parameter-map type inspect** *parameter-map-name* command takes precedence over the limit configured under the **parameter-map type inspect global** command.

The firewall considers Session Initiation Protocol (SIP) trunk traffic as a single session. However, the SIP trunk traffic contains a large number of application-layer gateway (ALG) flows of different users. When the throughput of the SIP trunk traffic is high compared to other traffic, the simultaneous packet limit causes packets to drop and users may experience call drops.

How to Configure the Number of Simultaneous Packets per Flow

Configuring Class Maps and Policy Maps for Simultaneous Packets per Flow

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode. • Enter your password if prompted.
Step 3	**class-map type inspect** {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any cmap-protocols`	Creates an inspect-type class map and enters class map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol tcp`	Configures the match criteria 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	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect policy1`	Creates an inspect-type policy map and enters policy map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect cmap-protocols`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Configures or modifies a policy for the default class.
Step 11	**end** **Example:** `Device(config-pmap)# end`	Exits policy map configuration mode and returns to privileged EXEC mode.

Configuring the Number of Simultaneous Packets per Flow

You can configure the number of simultaneous packets per flow after configuring either the **parameter-map type inspect** command or the **parameter-map type inspect global** command. The number of simultaneous packets per flow configured under the **parameter-map type inspect** command overwrites the number configured under the **parameter-map type inspect global** command.

You must configure the **session packet** command to configure the number of simultaneous packets per flow.

Note You must configure either Steps 3 and 4 or Steps 6 and 7.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **session packet** *number-of-simultaneous-packets*
5. **exit**

6. **parameter-map type inspect global**
7. **session packet** *number-of-simultaneous-packets*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode. • Enter your password if prompted.
Step 3	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect param1`	(Optional) Defines an inspect type parameter map, which configures connection thresholds, timeouts, and other parameters pertaining to the inspect action; and enters parameter-map type inspect configuration mode.
Step 4	**session packet** *number-of-simultaneous-packets* **Example:** `Device(config-profile)# session packet 55`	(Optional) Configures the number of simultaneous traffic packets that can be configured per session. • Valid values for the *number-of-simultaneous-packets* argument are 25 to 55.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and returns to global configuration mode.
Step 6	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	(Optional) Defines a global inspect parameter map and enters parameter-map type inspect configuration mode.
Step 7	**session packet** *number-of-simultaneous-packets* **Example:** `Device(config-profile)# session packet 35`	(Optional) Configures the number of simultaneous traffic packets that can be configured per session. • Valid values for the *number-of-simultaneous-packets* argument are 25 to 55.
Step 8	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Configuring Zones for Simultaneous Packets per Flow

This task shows how to configure security zones, a zone pair, and assign interfaces as zone members.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *security-zone*
4. **exit**
5. **zone security** *security-zone*
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
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** *security-zone* **Example:** `Device(config)# zone security z1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • You need two security zones to create a zone pair: a source zone and a destination zone.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** *security-zone* **Example:** `Device(config)# zone security z2`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • You need two security zones to create a zone pair: a source zone and a destination zone.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security zp-security source z1 destination z2`	Creates a zone pair and enters security zone pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect policy1`	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.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z1`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into 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 a part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/3`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Configurable Number of Simultaneous Packets per Flow

Example: Configuring Class Maps and Policy Maps for Simultaneous Packets per Flow

```
Device# configure terminal
Device(config)# class-map type inspect match-any cmap-protocols
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect policy1
Device(config-pmap)# class type inspect cmap-protocols
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap)# end
```

Example: Configuring the Number of Simultaneous Packets per Flow

You can configure the number of simultaneous packets per flow after configuring either the **parameter-map type inspect** command or the **parameter-map type inspect global** command. The number of simultaneous packets per flow configured under the **parameter-map type inspect** command overwrites the number configured under the **parameter-map type inspect global** command.

```
Device# configure terminal
Device(config)# parameter-map type inspect param1
Device(config-profile)# session packet 55
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# session packet 35
Device(config-profile)# end
```

Example: Configuring Zones for Simultaneous Packets per Flow

```
Device# configure terminal
Device(config)# zone security z1
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security zp-security source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect policy1
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# zone-member security z1
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/0/3
Device(config-if)# zone-member security z2
Device(config-if)# end
```

Additional References for Configurable Number of Simultaneous Packets per Flow

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

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

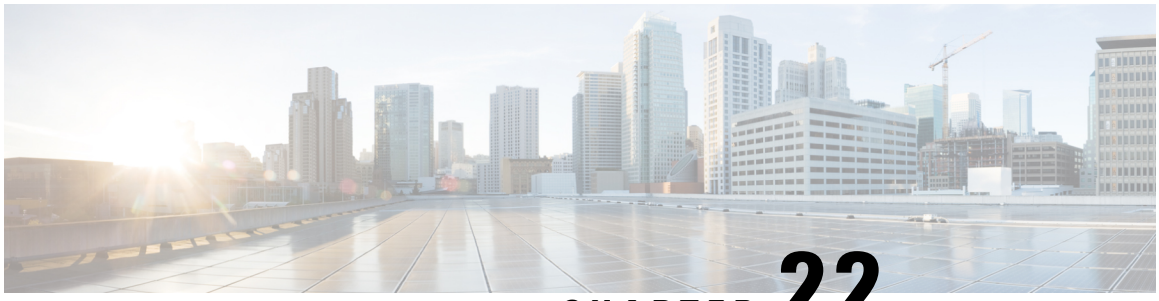
Feature Information for Configurable Number of Simultaneous Packets per Flow

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 Configurable Number of Simultaneous Packets per Flow

Feature Name	Releases	Feature Information
Configurable Number of Simultaneous Packets per Flow	Cisco IOS XE Release 3.11S	In zone-based policy firewalls, the number of simultaneous packets per flow was restricted to 25, and packets that exceeded the limit were dropped. The dropping of packets when the number is reached impacts network performance. The Configurable Number of Simultaneous Packets per Flow feature allows you to configure the number of simultaneous packets per flow from 25 to 100. In Cisco IOS XE Release 3.11S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers, the Cisco 4400 Series Integrated Services Routers, and the Cisco Cloud Services Routers 1000V Series. The following commands were introduced or modified: **session packet**, **show parameter-map type inspect**, **show platform hardware qfp feature firewall datapath scb**, **show platform hardware qfp feature firewall zone-pair**, and **show platform software firewall parameter-map**.



CHAPTER 22

LISP and Zone-Based Firewalls Integration and Interoperability

The LISP and Zone-Based Firewalls Integration and Interoperability feature enables inner-packet inspection of all Locator ID Separation Protocol (LISP) data packets that pass through a device. To enable LISP inner packet inspection, you have to configure the **lisp inner-packet inspection** command. Without LISP inner packet inspection, endpoint identifier (EID) devices in a LISP network will not have any firewall protection.

This module describes how to configure this 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.

Prerequisites for LISP and Zone-Based Firewall Integration and Interoperability

- The interchassis high availability configuration on active device and standby devices must be identical.

Restrictions for LISP and Zone-Based Firewall Integration and Interoperability

The following features are not supported:

- Locator ID Separator Protocol (LISP) mobility
- Zone-based firewall, LISP, and Web Cache Control Protocol (WCCP) interoperability

The following features are not supported when LISP inner packet inspection is enabled:

- Asymmetric routing
- LISP control message inspection
- LISP inner packet fragmentation
- Network Address Translation (NAT) and NAT 64
- TCP reset
- Virtual routing and forwarding (VRF)
- Virtual TCP (vTCP)
- VRF-Aware Software Infrastructure (VASI)
- Web Cache Communication Protocol (WCCP)

Information About LISP and Zone-Based Firewalls Integration and Interoperability

LISP Overview

The Locator ID Separation Protocol (LISP) is a network architecture and protocol. LISP replaces a single IP address with two numbering spaces—Routing Locators (RLOCs), which are topologically assigned to network attachment points and used for routing and forwarding of packets through the network; and Endpoint Identifiers (EIDs), which are assigned independently from the network topology and used for numbering devices, and are aggregated along administrative boundaries.

LISP defines functions for mapping between the two numbering spaces and encapsulating traffic originated by devices using non-routable EIDs for transport across a network infrastructure that routes and forwards using RLOCs. LISP provides a set of functions for devices to exchange information that is used to map non-routable EIDs to routable RLOCs.

LISP requires LISP-specific configuration of one or more LISP-related devices, such as the LISP egress tunnel router (ETR), ingress tunnel router (ITR), proxy ETR (PETR), proxy ITR (PITR), map resolver (MR), map server (MS), and LISP alternative logical topology (ALT) device.

Zone-Based Firewall and LISP Interoperability Overview

The zone-based firewall can be deployed either on the southbound or northbound of the Locator ID Separator Protocol (LISP) xTR device, depending on where the edge router (routers such as Cisco ASR 1000 Aggregation

Services Routers) is located in the network. The ingress tunnel router (ITR) and egress tunnel router (ETR) together are called the xTR device.

When the zone-based firewall is at the northbound of the xTR device; then the firewall can view LISP encapsulated packets, such as LISP tunneled packets, that pass through the network.

When the zone-based firewall is at the southbound of the xTR device, then the firewall can view the original packet. However; the firewall is not aware of any LISP xTR processing or do not see any LISP header. For egress packets, the xTR device does LISP encapsulation and adds the LISP header on top of the original packet after the firewall inspection. For ingress packets, the xTR device does LISP decapsulation (removal of the LISP header) before the firewall inspection and as a result, the firewall only inspects the original packet; and has no interaction with LISP at all.

This section describes the scenario when the zone-based firewall is deployed at the southbound of the LISP xTR device:

If an edge router is configured as a LISP xTR device to perform LISP encapsulation and decapsulation functions, you can configure the zone-based firewall between the LISP interface and the interfaces that face the LISP local endpoint identifier (EID) devices on the same edge router. LISP header decapsulation is performed before the header enters the zone-based firewall at the LISP interface. LISP header encapsulation is performed after the packet egresses from the firewall at the LISP interface. The firewall inspects only native traffic (what is native traffic here?) in the EID space.

This section describes the scenario when the zone-based firewall is deployed at the northbound of the LISP xTR devicce:

If more than one edge routers are deployed as load-sharing routers at the northbound of the xTR device, the firewall on the edge router is considered northbound of the xTR device. In this case, all packets that pass through the zone-based firewall are LISP encapsulated packets. When a packet arrives, the firewall inspects either the inner header or outer header of the LISP packets. By default, only the outer header is inspected. You can enable inner header inspection by using the **lisp inner-packet-inspection** command.

In Cisco IOS XE Release, if LISP inner packet inspection is enabled, the firewall only inspects the first fragmented inner packet, and all subsequent inner packets pass through the firewall without further inspection. If LISP inner packet inspection is enabled, the LISP instance ID is treated as virtual routing and forwarding (VRF) ID, and LISP packets that belong to different instance IDs are associated with different zone-based firewall sessions.

Feature Interoperability LISP

In Cisco IOS XE Release 3.13S, the LISP and Zone-Based Firewall Integration and Interoperability feature, works with the following features:

- IPv4 inner and outer headers
- IPv6 inner and outer headers
- LISP multitenancy
- Application layer gateways (ALGs)
- Application Inspection and Control (AIC)
- Mulitprotocol Label Switching (MPLS)
- In-Service Software Upgrade (ISSU)

- PxTR Case

Intrachassis and Interchassis High Availability for Zone-Based Firewall and LISP Integration

In Cisco IOS XE Release 3.14S, the LISP and Zone-Based Firewall Integration and Interoperability feature supports both intrachassis and interchassis high availability. When Location ID Separation Protocol (LISP) inner packet inspection is enabled, interchassis and intrachassis redundancy are supported at the xTR northbound device.

For LISP inner packet inspection at the northbound device, LISP instance ID is used as the virtual routing and forwarding (VRF) instance. The VRF configuration at northbound device is ignored if LISP inner packet inspection is enabled.

When two devices are located at the northbound of the xTR device and the xTR device is located inside the cloud, if LISP inner packet inspection is enabled on both devices, zone-based firewall sessions that are created for LISP inner packet flow is synced to the standby device.

A typical interchassis (box-to-box) high availability topology will have two devices in the routing locator (RLOC) space at the northbound of the xTR device. The xTR device sits in the inside network. If LISP inner packet inspection is enabled on both devices, zone-based firewall sessions that are created for LISP inner packets are synced to the standby device.

There are no configuration changes for intrachassis redundancy.

How to Configure LISP and Zone-Based Firewalls Integration and Interoperability

Enabling LISP Inner Packet Inspection

You can configure LISP inner packet inspection after configuring the **parameter-map type inspect global** command or the **parameter-map type inspect-global** command.

Note You cannot configure both these commands simultaneously.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **lisp inner-packet-inspection**
5. **end**
6. **show parameter-map type** {**inspect global** | **inspect-global**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device`	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 for connecting thresholds, timeouts, and other parameters pertaining to the inspect action, and enters parameter-map type inspect configuration mode.
Step 4	**lisp inner-packet-inspection** **Example:** `Device(config-profile)# lisp inner-packet-inspection`	Enables LISP inner packet inspection.
Step 5	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.
Step 6	**show parameter-map type** {**inspect global** | **inspect-global**} **Example:** `Device# show parameter-map type inspect-global`	Displays global inspect-type parameter map information.

Example

The following sample output from the **show parameter-map type inspect-global** command displays that LISP inner-packet inspection is enabled:

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

parameter-map type inspect-global
  log dropped-packet off
  alert on
  aggressive aging disabled
  syn_flood_limit  unlimited
  tcp window scaling enforcement loose off
  max incomplete unlimited  aggressive aging disabled
  max_incomplete TCP unlimited
  max_incomplete UDP unlimited
  max_incomplete ICMP unlimited
  application-inspect all
  vrf default inspect vrf-default
  vrf vrf2 inspect vrf-default
  vrf vrf3 inspect vrf-default
```

```
lisp inner-packet-inspection
```

Configuring Interchassis High Availability for LISP Inner Packet Inspection

Configuring the xTR Southbound Interface for Interchassis High Availability

Before you begin

Prerequisites

- Zones and zone-pairs must be configured.
- Redundancy and redundancy groups must be configured. See, the "Configuring Firewall Stateful Interchassis Redundancy" module in the *Zone-Based Policy Firewall Configuration Guide* for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **description** *string*
6. **ip address** *ip-address mask*
7. **exit**
8. **interface** *type number*
9. **description** *string*
10. **zone-member security** *zone-name*
11. **exit**
12. **interface** *type number*
13. **description** *string*
14. **ip address** *ip-address mask*
15. **zone-member security** *zone-name*
16. **cdp enable**
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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet 1/3/0`	Configures an interface and enters interface configuration mode.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding lower`	Associates a VRF instance or a virtual network with an interface or subinterface.
Step 5	**description** *string* **Example:** `Device(config-if)# description facing RLOC and the LISP cloud; has a LISP header.`	Adds a description to an interface configuration. • The zone-based firewall cannot be configured at this interface.
Step 6	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.1.27 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface LISP 0`	Configures an interface and enters interface configuration mode. • This is the LISP virtual interface.

Step 9	**description** *string*
	Example:
	`Device(config-if)# description LISP virtual interface.`
	`Adds LISP header after firewall inspection or removes LISP header before firewall inspection.`
		Adds a description to an interface configuration.

Step 10	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security ge0-0-3a`	Attaches an interface to a security zone.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**interface** *type number* **Example:** `Device(config)# interface tengigabitethernet 0/3/0`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 13	**description** *string* **Example:** `Device(config-if)# description facing internal network, does not have a LISP header.`	Adds a description to an interface configuration.
Step 14	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.2.5 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 15	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security ge0-0-0`	Attaches an interface to a security zone.
Step 16	**cdp enable** **Example:** `Device(config-if)# cdp enable`	Enable Cisco Discovery Protocol (CDP) on an interface.
Step 17	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the xTR Northbound Interface for LISP Inner Packet Inspection

In this configuration, a Locator ID Separation Protocol (LISP) virtual interface is not needed because at northbound the LISP header is not inspected. However, you can configure the zone-based firewall to inspect either LISP inner packets or outer packets.

Before you begin

- Zones and zone-pairs must be configured.
- Redundancy and redundancy groups must be configured. See, the "Configuring Firewall Stateful Interchassis Redundancy" module in the *Zone-Based Policy Firewall Configuration Guide* for more information.

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. **negotiation auto**
8. **redundancy rii** *id*

9. **redundancy group** *id* **ip** *virtual-ip* **exclusive decrement** *value*
10. **exit**
11. **interface** *type number*
12. **description** *string*
13. **ip address** *ip-address mask*
14. **zone-member security** *zone-name*
15. **negotiation auto**
16. **redundancy rii** *id*
17. **redundancy group** *id* **ip** *virtual-ip* **exclusive decrement** *value*
18. **ip virtual-reassembly**
19. **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 1/2/1`	Configures an interface and enters interface configuration mode. • This interface can see the entire LISP packet.
Step 4	**description** *string* **Example:** `Device(config-if)# description RLOC-space/north LAN`	Adds a description to an interface configuration.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 198.51.100.8 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security ge0-0-3`	Attaches an interface to a security zone.
Step 7	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables advertisement of speed, duplex mode, and flow control on a Gigabit Ethernet interface.

	Command or Action	Purpose
Step 8	**redundancy rii** *id* **Example:** `Device(config-subif)# redundancy rii 200`	Configures the redundancy interface identifier (RII) for redundancy group protected traffic interfaces
Step 9	**redundancy group** *id* **ip** *virtual-ip* **exclusive decrement** *value* **Example:** `Device(config-if)# redundancy group 1 ip 198.51.100.12 exclusive decrement 50`	Enables the redundancy group (RG) traffic interface configuration.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/3`	Configures an interface and enters interface configuration mode. • This interface can see the entire LISP packet.
Step 12	**description** *string* **Example:** `Device(config-if)# description RLOC-space/south LAN`	Adds a description to an interface configuration.
Step 13	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 198.51.100.27 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security ge0-0-0`	Attaches an interface to a security zone.
Step 15	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables advertisement of speed, duplex mode, and flow control on a Gigabit Ethernet interface.
Step 16	**redundancy rii** *id* **Example:** `Device(config-subif)# redundancy rii 300`	Configures the redundancy interface identifier (RII) for redundancy group protected traffic interfaces
Step 17	**redundancy group** *id* **ip** *virtual-ip* **exclusive decrement** *value* **Example:** `Device(config-if)# redundancy group 1 ip 194.88.4.1 exclusive decrement 50`	Enables the RG traffic interface configuration.

	Command or Action	Purpose
Step 18	**ip virtual-reassembly** **Example:** `Device(config-if)# ip virtual-reassembly`	Enables virtual fragment reassembly (VFR) on an interface.
Step 19	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for LISP and Zone-Based Firewalls Integration and Interoperability

Example: Enbaling LISP Inner Packet Inspection

```
Device# configure terminal
Device(config)# parameter-map type inspect-global
Device(config-profile)# lisp inner-packet-inspection
Device(config-profile)# end
```

The following example shows a zone-based firewall configuration with LISP inner-packet inspection enabled:

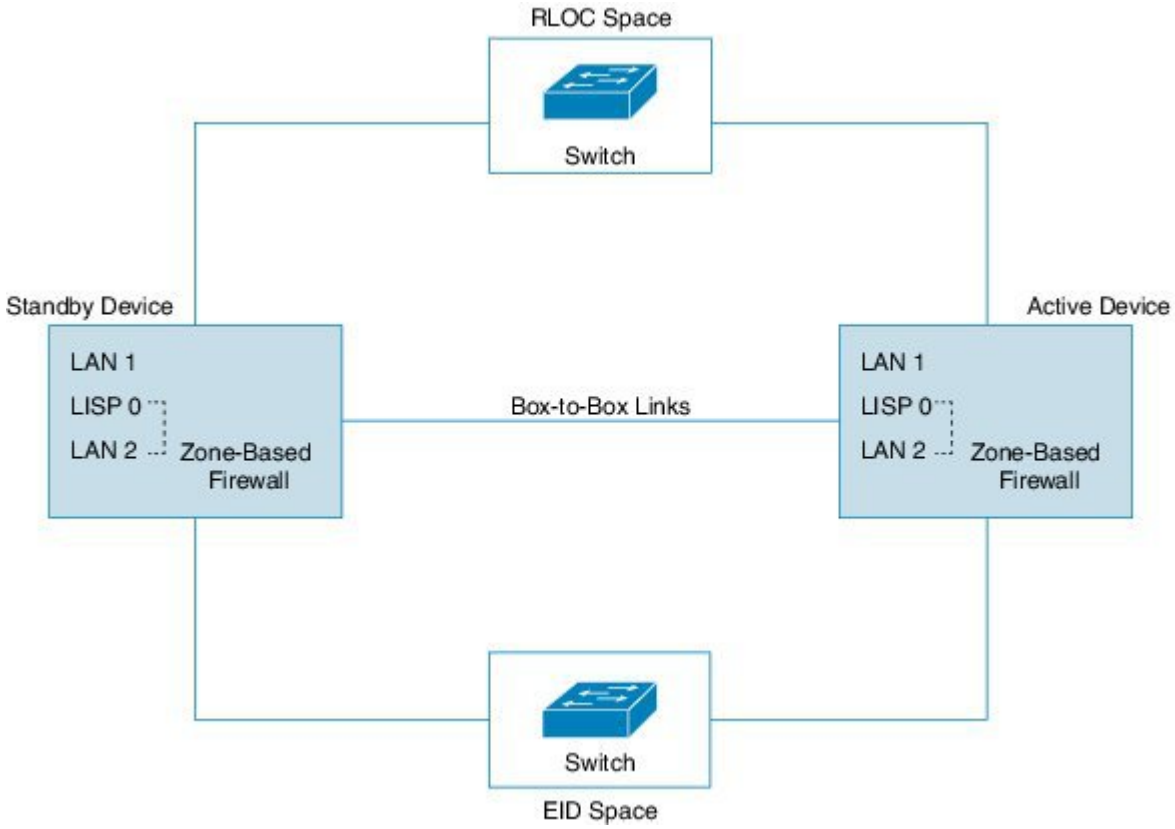
```
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family

class-map type inspect match-any c-ftp-tcp
 match protocol ftp
 match protocol telnet
 match protocol http
 match protocol tcp
 match protocol udp
!
policy-map type inspect p1
 class type inspect c-ftp-tcp
  inspect
 class class-default
!
zone security ge0-0-0
!
zone security ge0-0-3
!
zone-pair security zp-ge000-ge003 source ge0-0-0 destination ge0-0-3
 service-policy type inspect p1
!
zone-pair security zp-ge003-ge000 source ge0-0-3 destination ge0-0-0
 service-policy type inspect p1
!
interface TenGigabitEthernet 1/3/0
```

```
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:100::2/64
 zone-member security ge0-0-0
!
interface TenGigabitEthernet 0/3/0
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:200::2/64
 zone-member security ge0-0-3
!
parameter-map type inspect global
 lisp inner-packet-inspection
 log dropped-packet off
  alert on
!
```

Example: Configuring Interchassis High Availability for LISP Inner Packet Inspection

In the figure below, LISP 0 is the LISP virtual interface and this interface performs LISP header encapsulation and decapsulation. Firewall zone pairs must be configured between the LISP 0 interface and LAN2. Redundant Groups (RGs) are configured on both LAN1 and LAN2. RGs configured under LAN2 is used to synchronize zone-based firewall sessions between active and standby devices.

Figure 28: xTR Devices with Box-to-Box High Availability Deployment



The following is a sample interchassis high availability configuration with a LISP virtual interface:

```
! Configuration on Device 1:
Device(config)# redundancy
Device(config-red)# application
Device(config-red-app)# group 1
Device(config-red-app-grp)# name RG1
Device(config-red-app-grp)# priority 205 failover-threshold 200
Device(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1
Device(config-red-app-grp)# data gigabitethernet 0/0/2
!
!
Device(config)# parameter-map type inspect global
Device(config-profile)# redundancy
Device(config-profile)# redundancy delay 10
Device(config-profile)# lisp inner-packet-inspection
Device(config-profile)# log dropped-packet off
Device(config-profile)# alert on
!
!
Device(config)# class-map type inspect match-all ha-class
Device(config-cmap)# match protocol tcp
!
Device(config)# class-map type inspect match-any cmap-any
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol ftp
Device(config-cmap)# match protocol icmp
!
Device(config)# policy-map type inspect ha-policy
Device(config-pmap)# class type inspect ha-class
Device(config-pmap-c)# inspect
!
Device(config-pmap)# class class-default
Device(config-pmap)# drop
!
Device(config)# policy-map type inspect pmap-ha
Device(config-pmap)# class type inspect cmap-any
Device(config-pmap-c)# inspect
!
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
!
Device(config)# zone security ge0-0-3a
!
Device(config)# zone security ge0-0-0a
!
Device(config)# zone-pair security ha-in-out source ge0-0-3a destination ge0-0-0a
Device(config-sec-zone-pair)# service-policy type inspect ha-policy
!
Device(config)# zone-pair security ha-out-in source ge0-0-0a destination ge0-0-3a
Device(config-sec-zone-pair)# service-policy type inspect pmap-ha
!
Device(config)# ip vrf lower
!
Device(config)# interface TenGigabitEthernet 1/3/0
Device(config-if)# vrf forwarding lower
Device(config-if)# description RLOC-space/north LAN ! This interface can see LISP packets.
Device(config-if)# ip address 192.0.1.27 255.255.255.0
!
Device(config)# interface LISP 0 ! The LISP virtual interface.
This interface decapsulates/encapsulates the LISP header.
Device(config-if)# zone-member security ge0-0-3a
Device(config-if)# redundancy rii 13
!
Device(config)# interface TenGigabitEthernet 0/3/0
```

```
Device(config-if)# vrf forwarding lower
Device(config-if)# description EID_space/south LAN ! This interface only sees native packet.

The LISP header is removed by the LISP virtual interface.
Device(config-if)# zone-member security ge0_0_0a
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# redundancy rii 10
Device(config-if)# redundancy group 2 ip 192.0.2.3 exclusive decrement 50
!


! Configuration on Device 2:
Device(config)# redundancy
Device(config-red)# application
Device(config-red-app)# group 1
Device(config-red-app-grp)# name RG1
Device(config-red-app-grp)# priority 195 failover-threshold 190
Device(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1
Device(config-red-app-grp)# data gigabitethernet 0/0/2
!
!
Device(config)# parameter-map type inspect global
Device(config-profile)# redundancy
Device(config-profile)# redundancy delay 10
Device(config-profile)# lisp inner-packet-inspection
Device(config-profile)# log dropped-packet off
Device(config-profile)# alert on
!
Device(config)# class-map type inspect match-all ha-class
Device(config-cmap)# match protocol tcp
!
Device(config)# class-map type inspect match-any cmap-any
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol ftp
Device(config-cmap)# match protocol icmp
!
Device(config)# policy-map type inspect ha-policy
Device(config-pmap)# class type inspect ha-class
Device(config-pamp-c)# inspect
!
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
!
Device(config)# policy-map type inspect pmap-ha
Device(config-pmap)# class type inspect cmap-any
Device(config-pmap-c)# inspect
!
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
!
Device(config)# zone security ge0-0-3a
!
Device(config)# zone security ge0-0-0a
!
Device(config)# zone-pair security ha-in-out source ge0-0-3a destination ge0-0-0a
Device(config-sec-zone-pair)# service-policy type inspect ha-policy
!
Device(config)# zone-pair security ha-in-out source ge0-0-0a destination ge0-0-3a
Device(config-sec-zone-pair)# service-policy type inspect pmap-ha
!
Device(config)# ip vrf lower
!
Device(config)# interface TenGigabitEthernet 1/3/0
```

```
Device(config-if)# vrf forwarding lower
Device(config-if)# description RLOC-space/north LAN ! This interface can see LISP packets.
Device(config-if)# ip address 192.0.1.32 255.255.255.0
!
Device(config)# interface LISP 0 ! The LISP virtual interface.
This interface decapsulates/encapsulates the LISP header.
Device(config-if)# zone-member security ge0-0-3a
Device(config-if)# redundancy rii 13
!
Device(config)# interface TenGigabitEthernet 0/3/0
Device(config-if)# vrf forwarding lower
Device(config-if)# description EID_space/south LAN !This interface only sees native packet.

The LISP header is removed by the LISP virtual interface.>>>>
Device(config-if)# zone-member security ge0-0-0a
Device(config-if)# ip address 192.0.2.5 255.255.255.0
Device(config-if)# redundancy rii 10
Device(config-if)# redundancy group 2 ip 192.0.2.7 exclusive decrement 50
!
```

Additional References for LISP and Zone-Based Firewalls Integration and Interoperability

Related Documents

Related Topic	Document Title
Cisco commands	Master Command List, All Releases
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
LISP commands	Cisco IOS IP Routing: LISP Command Reference
LISP configuration guide	*IP Routing: LISP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 6830	*The Locator/ID Separation Protocol (LISP)*

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

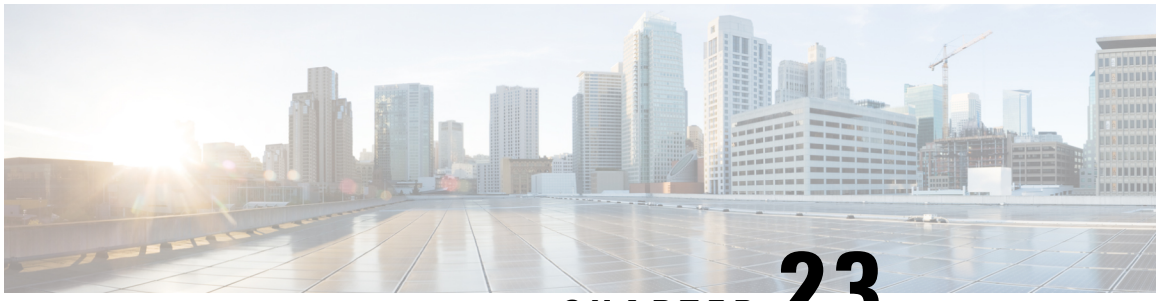
Feature Information for LISP and Zone-Based Firewall Integration and Interoperability

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 LISP and Zone-Based Firewall Integration and Interoperability

Feature Name	Releases	Feature Information
LISP and Zone-Based Firewall Integration and Interoperability	Cisco IOS XE Release 3.13S	The LISP and Zone-Based Firewalls Integration and Interoperability feature enables inner-packet inspection of all Locator ID Separation Protocol (LISP) data packets that pass through a device. To enable LISP inner packet inspection, you have to configure the lisp inner-packet inspection command. Without LISP inner inspection, endpoint identifier (EID) devices in a LISP network will not have any firewall protection. The following commands were introduced or modified by this feature: **lisp inner-packet-inspection**, **show parameter-map type inspect-global**, and **show parameter-map type inspect global**.
Intrachassis and Interchassis High Availability for Zone-Based Firewall and LISP Integration	Cisco IOS XE Release 3.14S	In Cisco IOS XE Release 3.14S, the LISP and Zone-Based Firewall Integration and Interoperability feature supports both intrachassis and interchassis high availability. No commands were introduced or modified by this feature.



CHAPTER 23

Firewall High-Speed Logging

The Firewall High-Speed Logging feature supports the high-speed logging (HSL) of firewall messages by using NetFlow Version 9 as the export format.

This module describes how to configure HSL for zone-based policy 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 Firewall High-Speed Logging

Firewall High-Speed Logging Overview

Zone-based firewalls support high-speed logging (HSL). When HSL is configured, a firewall provides a log of packets that flow through routing devices (similar to the NetFlow Version 9 records) to an external collector. Records are sent when sessions are created and destroyed. Session records contain the full 5-tuple information (the source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements.

HSL allows a firewall to log records with minimum impact to packet processing. The firewall uses buffered mode for HSL. In buffered mode, a firewall logs records directly to the high-speed logger buffer, and exports of packets separately.

A firewall logs the following types of events:

- Audit—Session creation and removal notifications.
- Alert—Half-open and maximum-open TCP session notifications.
- Drop—Packet-drop notifications.
- Pass—Packet-pass (based on the configured rate limit) notifications.
- Summary—Policy-drop and pass-summary notifications.

The NetFlow collector issues the **show platform software interface F0 brief** command to map the FW_SRC_INTF_ID and FW_DST_INTF_ID interface IDs to the interface name.

The following sample output from the **show platform software interface F0 brief** command shows that the ID column maps the interface ID to the interface name (Name column):

```
Device# show platform software interface F0 brief

Name                            ID      QFP ID
GigabitEthernet0/2/0            16         9
GigabitEthernet0/2/1            17        10
GigabitEthernet0/2/2            18        11
GigabitEthernet0/2/3            19        12
```

NetFlow Field ID Descriptions

The following table lists NetFlow field IDs used within the firewall NetFlow templates:

Table 28: NetFlow Field IDs

Field ID	Type	Length	Description
NetFlow ID Fields (Layer 3 IPv4)			
FW_SRC_ADDR_IPV4	8	4	Source IPv4 address
FW_DST_ADDR_IPV4	12	4	Destination IPv4 address
FW_SRC_ADDR_IPV6	27	16	Source IPv6 address
FW_DST_ADDR_IPV6	28	16	Destination IPv6 address
FW_PROTOCOL	4	1	IP protocol value
FW_IPV4_IDENT	54	4	IPv4 identification
FW_IP_PROTOCOL_VERSION	60	1	IP protocol version
Flow ID Fields (Layer 4)			
FW_TCP_FLAGS	6	1	TCP flags
FW_SRC_PORT	7	2	Source port
FW_DST_PORT	11	2	Destination port
FW_ICMP_TYPE	176	1	ICMP [1] type value

Field ID	Type	Length	Description
FW_ICMP_CODE	177	1	ICMP code value
FW_ICMP_IPV6_TYPE	178	1	ICMP Version 6 (ICMPv6) type value
FW_ICMP_IPV6_CODE	179	1	ICMPv6 code value
FW_TCP_SEQ	184	4	TCP sequence number
FW_TCP_ACK	185	4	TCP acknowledgment number
Flow ID Fields (Layer 7)			
FW_L7_PROTOCOL_ID	95	2	Layer 7 protocol ID. Identifies the Layer 7 application classification used by firewall inspection. Normal records use 2 bytes, but optional records use 4 bytes.
Flow Name Fields (Layer 7)			
FLOW_FIELD_L7_PROTOCOL_NAME	96	32	Layer 7 protocol name. Identifies the Layer 7 protocol name that corresponds to the Layer 7 protocol ID (FW_L7_PROTOCOL_ID).
Flow ID Fields (Interface)			
FW_SRC_INTF_ID	10	2	Ingress SNMP [2] ifIndex
FW_DST_INTF_ID	14	2	Egress SNMP ifIndex
FW_SRC_VRF_ID	234	4	Ingress (initiator) VRF [3] ID
FW_DST_VRF_ID	235	4	Egress (responder) VRF ID
FW_VRF_NAME	236	32	VRF name
Mapped Flow ID Fields (Network Address Translation)			
FW_XLATE_SRC_ADDR_IPV4	225	4	Mapped source IPv4 address
FW_XLATE_DST_ADDR_IPV4	226	4	Mapped destination IPv4 address
FW_XLATE_SRC_PORT	227	2	Mapped source port
FW_XLATE_DST_PORT	228	2	Mapped destination port
Status and Event Fields			

Field ID	Type	Length	Description
FW_EVENT	233	1	High level event codes • 0—Ignore (invalid) • 1—Flow created • 2—Flow deleted • 3—Flow denied • 4—Flow alert
FW_EXT_EVENT	35,001	2	Extended event code. For normal records the length is 2 byte, and 4 byte for optional records.
Timestamp and Statistics Fields			
FW_EVENT_TIME_MSEC	323	8	Time, in milliseconds, (time since 0000 hours UTC [4] January 1, 1970) when the event occurred (if the event is a microevent, use 324 and 325, if it is a nanoevent)
FW_INITIATOR_OCTETS	231	4	Total number of Layer 4 payload bytes in the packet flow that arrives from the initiator
FW_RESPONDER_OCTETS	232	4	Total number of Layer 4 payload bytes in the packet flow that arrives from the responder
AAA Fields			
FW_USERNAME	40,000	20 or 64 depending on the template	AAA [5] user name
FW_USERNAME_MAX	40,000	64	AAA user name of the maximum permitted size
Alert Fields			
FW_HALFOPEN_CNT	35,012	4	Half-open session entry count
FW_BLACKOUT_SECS	35,004	4	Time, in seconds, when the destination is blacked out or unavailable
FW_HALFOPEN_HIGH	35,005	4	Configured maximum rate of TCP half-open session entries logged in one minute

Field ID	Type	Length	Description
FW_HALFOPEN_RATE	35,006	4	Current rate of TCP half-open session entries logged in one minute
FW_MAX_SESSIONS	35,008	4	Maximum number of sessions allowed for this zone pair or class ID
Miscellaneous			
FW_ZONEPAIR_ID	35,007	4	Zone pair ID
FW_CLASS_ID	51	4	Class ID
FW_ZONEPAIR_NAME	35,009	64	Zone pair name
FW_CLASS_NAME	100	64	Class name
FW_EXT_EVENT_DESC	35,010	32	Extended event description
FLOW_FIELD_CTS_SRC_GROUP_TAG	34000	2	Cisco Trustsec source tag
FW_SUMMARY_PKT_CNT	35,011	4	Number of packets represented by the drop/pass summary record
FW_EVENT_LEVEL	33003	4	Defines the level of the logged event • 0x01—Per box • 0x02—VRF • 0x03—Zone • 0x04—Class map • Other values are undefined
FW_EVENT_LEVEL_ID	33,004	4	Defines the identifier for the FW_EVENT_LEVEL field • If FW_EVENT_LEVEL is 0x02 (VRF), this field represents VRF_ID. • If FW_EVENT_LEVEL is 0x03 (zone), this field represents ZONE_ID. • If FW_EVENT_LEVEL is 0x04 (class map), this field represents CLASS_ID. • In all other cases the field ID will be 0 (zero). If FW_EVENT_LEVEL is not present, the value of this field must be zero.

Field ID	Type	Length	Description
FW_CONFIGURED_VALUE	33,005	4	Value that represents the configured half-open, aggressive-aging, and event-rate monitoring limit. The interpretation of this field value depends on the associated FW_EXT_EVENT field.
FW_ERM_EXT_EVENT	33,006	2	Extended event-rate monitoring code
FW_ERM_EXT_EVENT_DESC	33,007	N (string)	Extended event-rate monitoring event description string

[1] Internet Control Message Protocol
[2] Simple Network Management Protocol
[3] virtual routing and forwarding
[4] Coordinated Universal Time
[5] Authentication, Authorization, and Accounting

HSL Messages

The following are sample syslog messages from an Cisco ASR 1000 Series Aggregation Services Router:

Table 29: Syslog Messages and Their Templates

Message Identifier	Message Description	HSL Template
FW-6-DROP_PKT Type: Info	Dropping %s pkt from %s %CA:%u => %CA:%u (target:class)-(%s:%s) %s %s with ip ident %u %s %s Explanation: Packet dropped by firewall inspection. %s: tcp/udp/icmp/unknown prot/L7 prot %s:interface %CA:%u ip/ip6 addr: port %s:%s: zone pair name/ class name %s "due to" %s: fw_ext_event name %u ip ident %s: if tcp, tcp seq/ack number and tcp flags %s: username	FW_TEMPLATE_DROP_V4 or FW_TEMPLATE_DROP_V6

Message Identifier	Message Description	HSL Template
FW-6-SESS_AUDIT_TRAIL_START Type: Info	(target:class)-(%s:%s):Start %s session: initiator (%CA:%u) -- responder (%CA:%u) from %s %s %s Explanation: Start of an inspection session. This message is issued at the start of each inspection session and it records the source/destination addresses and ports. %s:%s: zonepair name: class name %s: l4/l7 protocolname %CA:%u ip/ip6 addr: port %s : interface %s : username %s : TODO Actual log: *Jan 21 20:13:01.078: %IOSXE-6-PLATFORM: F0: cpp_cp: CPP:00 Thread:125 TS:00000010570290947309 %FW-6-SESS_AUDIT_TRAIL_START: Start tcp session: initiator (10.1.1.1:43365) -- responder (10.3.21.1:23) from FastEthernet0/1/0	FW_TEMPLATE_START_AUDIT_V4 or FW_TEMPLATE_START_AUDIT_V6

Message Identifier	Message Description	HSL Template
FW-6-SESS_AUDIT_TRAIL Type: Info	(target:class)-(%s:%s):Stop %s session: initiator (%CA:%u) sent %u bytes -- responder (%CA:%u) sent %u bytes , from %s %s Explanation: Per-session transaction log of network activities. This message is issued at the end of each inspection session, and it records the source/destination addresses and ports, and the number of bytes transmitted by the client and the server. %s:%s: zonepair name: class name %s: l4/l7 protocolname %CA:%u ip/ip6 addr: port %u bytes counters %s: interface %s : TODO Actual log: *Jan 21 20:13:15.889: %IOSXE-6-PLATFORM: F0: cpp_cp: CPP:00 Thread:036 TS:00000010585102587819 %FW-6-SESS_AUDIT_TRAIL: Stop tcp session: initiator (10.1.1.1:43365) sent 35 bytes -- responder (11.1.1.1:23) sent 95 bytes, from FastEthernet0/1/0	FW_TEMPLATE_STOP_AUDIT_V4 or FW_TEMPLATE_STOP_AUDIT_V6
FW-4-UNBLOCK_HOST Type: Warning	(target:class)-(%s:%s):New TCP connections to host %CA no longer blocked Explanation: New TCP connection attempts to the specified host are no longer blocked. This message indicates that the blocking of new TCP connection attempts to the specified host has been removed. %s:%s: zonepair name: class name %CA: ip/ip6 addr	FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V4 or FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V6 with fw_ext_event id: FW_EXT_ALERT_UNBLOCK_HOST

Message Identifier	Message Description	HSL Template
FW4HOST_TCP_ALERT_ON Type: Warning	"(target:class)-(%s:%s):Max tcp half-open connections (%u) exceeded for host %CA. Explanation: Exceeded the max-incomplete host limit for half-open TCP connections. This message indicates that a high number of half-open connections is coming to a protected server, and this may indicate that a SYN flood attack is in progress. %s:%s: zonepair name: class name %u: half open cnt %CA: ip/ip6 addr	FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V4 or FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V6 with fw_ext_event id: FW_EXT_ALERT_HOST_TCP_ALERT_ON
FW-2-BLOCK_HOST Type: Critical	(target:class)-(%s:%s):Blocking new TCP connections to host %CA for %u minute%s (half-open count %u exceeded). Explanation: Exceeded the max-incomplete host threshold for TCP connections. Any subsequent new TCP connection attempts to the specified host is denied, and the blocking option is configured to block all subsequent new connections. The blocking will be removed when the configured block time expires. %s:%s: zonepair name: class name %CA: ip/ip6 addr %u blackout min %s: s if > 1 min blackout time %u: half open counter	FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V4 or FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V6 with fw_ext_event id: FW_EXT_ALERT_BLOCK_HOST

Message Identifier	Message Description	HSL Template
FW-4-ALERT_ON Type: Warning	(target:class)-(%s:%s):%s, count (%u/%u) current rate: %u Explanation : Either the max-incomplete high threshold of half-open connections or the new connection initiation rate has been exceeded. This error message indicates that an unusually high rate of new connections is coming through the firewall, and a DOS attack may be in progress. This message is issued only when the max-incomplete high threshold is crossed. %s:%s: zonepair name: class name %s: "getting aggressive" %u/%u halfopen cnt/high %u: current rate	FW_TEMPLATE_ALERT_HALFOPEN_V4 or FW_TEMPLATE_ALERT_HALFOPEN_V6: with fw_ext_event id FW_EXT_SESS_RATE_ALERT_ON
FW-4-ALERT_OFF Type: Warning	(target:class)-(%s:%s):%s, count (%u/%u) current rate: %u Explanation: Either the number of half-open connections or the new connection initiation rate has gone below the max-incomplete low threshold. This message indicates that the rate of incoming new connections has slowed down and new connections are issued only when the max-incomplete low threshold is crossed. %s:%s: zonepair name: class name %s: "calming down" %u/%u halfopen cnt/high %u: current rate	FW_TEMPLATE_ALERT_HALFOPEN_V4 or FW_TEMPLATE_ALERT_HALFOPEN_V6: with fw_ext_event id FW_EXT_SESS_RATE_ALERT_OFF

Message Identifier	Message Description	HSL Template
FW-4-SESSIONS_MAXIMUM Type: Warning	Number of sessions for the firewall policy on "(target:class)-(%s:%s) exceeds the configured sessions maximum value %u Explanation: The number of established sessions have crossed the configured sessions maximum limit. %s:%s: zonepair name: class name %u: max session	FW_TEMPLATE_ALERT_MAX_SESSION
FW-6-PASS_PKT Type: Info	Passing %s pkt from %s %CA:%u => %CA:%u (target:class)-(%s:%s) %s %s with ip ident %u Explanation: Packet is passed by firewall inspection. %s: tcp/udp/icmp/unknown prot %s:interface %CA:%u src ip/ip6 addr: port %CA:%u dst ip/ip6 addr: port %s:%s: zonepair name: class name %s %s: "due to", "PASS action found in policy-map" %u: ip ident	FW_TEMPLATE_PASS_V4 or FW_TEMPLATE_PASS_V6
FW-6-LOG_SUMMARY Type: Info	%u packet%s %s from %s %CA:%u => %CA:%u (target:class)-(%s:%s) %s Explanation : Log summary for the number of packets dropped/passed %u %s: pkt_cnt, "s were" or "was" %s: "dropped"/ "passed" %s: interface %CA:%u src ip/ip6 addr: port %CA:%u dst ip/ip6 addr: port %s:%s: zonepair name: class name %s: username	FW_TEMPLATE_SUMMARY_V4 or FW_TEMPLATE_SUMMARY_V6 with FW_EVENT: 3 - drop 4 - pass

Firewall Extended Events

The event name of the firewall extended event maps the firewall extended event value to an event ID. Use the event name option record to obtain the mapping between an event value and an event ID.

Extended events are not part of standard firewall events (inspect, pass, or drop).

The following table describes the firewall extended events applicable prior to Cisco IOS XE Release 3.9S.

Table 30: Firewall Extended Events and Event Descriptions for Releases earlier than Cisco IOS XE Release 3.9S

Value	Event ID	Description
0	FW_EXT_LOG_NONE	No specific extended event.
1	FW_EXT_ALERT_UNBLOCK_HOST	New TCP connection attempts to the specified host are no longer blocked.
2	FW_EXT_ALERT_HOST_TCP_ALERT_ON	Maximum incomplete host limit for half-open TCP connections are exceeded.
3	FW_EXT_ALERT_BLOCK_HOST	All subsequent new TCP connection attempts to the specified host are denied because the maximum incomplete host threshold of half-open TCP connections is exceeded, and the blocking option is configured to block subsequent new connections.
4	FW_EXT_SESS_RATE_ALERT_ON	Maximum incomplete high threshold of half-open connections is exceeded, or the new connection initiation rate is exceeded.
5	FW_EXT_SESS_RATE_ALERT_OFF	Number of half-open TCP connections is below the maximum incomplete low threshold, or the new connection initiation rate has gone below the maximum incomplete low threshold.
6	FW_EXT_RESET	Reset connection.
7	FW_EXT_DROP	Drop connection.
10	FW_EXT_L4_NO_NEW_SESSION	No new session is allowed.
12	FW_EXT_L4_INVALID_SEG	Invalid TCP segment.
13	FW_EXT_L4_INVALID_SEQ	Invalid TCP sequence number.
14	FW_EXT_L4_INVALID_ACK	Invalid TCP acknowledgment (ACK).
15	FW_EXT_L4_INVALID_FLAGS	Invalid TCP flags.
16	FW_EXT_L4_INVALID_CHKSM	Invalid TCP checksum.
18	FW_EXT_L4_INVALID_WINDOW_SCALE	Invalid TCP window scale.

Value	Event ID	Description
19	FW_EXT_L4_INVALID_TCP_OPTIONS	Invalid TCP options.
20	FW_EXT_L4_INVALID_HDR	Invalid Layer 4 header.
21	FW_EXT_L4_OOO_INVALID_SEG	OoO [6] invalid segment.
24	FW_EXT_L4_SYNFLOOD_DROP	Synchronized (SYN) flood packets are dropped.
25	FW_EXT_L4_SCB_CLOSED	Session is closed while receiving packets.
26	FW_EXT_L4_INTERNAL_ERR	Firewall internal error.
27	FW_EXT_L4_OOO_SEG	OoO segment.
28	FW_EXT_L4_RETRANS_INVALID_FLAGS	Invalid retransmitted packet.
29	FW_EXT_L4_SYN_IN_WIN	Invalid SYN flag.
30	FW_EXT_L4_RST_IN_WIN	Invalid reset (RST) flag.
31	FW_EXT_L4_STRAY_SEG	Stray TCP segment.
32	FW_EXT_L4_RST_TO_RESP	Sending reset message to the responder.
33	FW_EXT_L4_CLOSE_SCB	Closing a session.
34	FW_EXT_L4_ICMP_INVAL_RET	Invalid ICMP [7] packet.
37	FW_EXT_L4_MAX_HALFSESSION	Maximum half-open session limit is exceeded.
38	FW_EXT_NO_RESOURCE	Resources (memory) are not available.
40	FW_EXT_INVALID_ZONE	Invalid zone.
41	FW_EXT_NO_ZONE_PAIR	Zone pairs are not available.
42	FW_EXT_NO_TRAFFIC_ALLOWED	Traffic is not allowed.
43	FW_EXT_FRAGMENT	Packet fragments are dropped.
44	FW_EXT_PAM_DROP	PAM [8] action is dropped.
45	FW_EXT_NOT_INITIATOR	Not a session-initiating packet. Occurs due to one of the following reasons: • If the protocol is TCP, the first packet is not a SYN packet. • If the protocol is ICMP, the first packet is not an ECHO or a TIMESTAMP packet.

Value	Event ID	Description
48	FW_EXT_ICMP_ERROR_PKTS_BURST	ICMP error packets came in burst mode. In burst mode, packets are sent repeatedly without waiting for a response from the responder interface.
49	FW_EXT_ICMP_ERROR_MULTIPLE_UNREACH	More than one ICMP error of type “destination unreachable” is received.
50	FW_EXT_ICMP_ERROR_L4_INVALID_SEQ	Embedded packet in the ICMP error message has an invalid sequence number.
51	FW_EXT_ICMP_ERROR_L4_INVALID_ACK	Embedded packet in the ICMP error message has an invalid acknowledge (ACK) number.
52	FW_EXT_MAX	Never used.

[6] Out-of-Order
[7] Internet Control Message Protocol
[8] Port-to-Application Mapping

The following table describes the firewall extended events from that are applicable to Cisco IOS XE Release 3.9S and later releases.

Table 31: Firewall Extended Events and Event Descriptions for Cisco IOS XE Release 3.9S and Later Releases

Value	Event ID	Description
0	FW_EXT_LOG_NONE	No specific extended event.
1	FW_EXT_FW_DROP_L4_TYPE_INVALID_HDR	Small datagram that cannot contain the Layer 4 ICMP, TCP, or UDP headers.
2	FW_EXT_FW_DROP_L4_TYPE_INVALID_ACK_FLAG	Did not contain an ACK flag, or a RST flag was set in the SYN/ACK packet during the TCP three-way handshake and the packet had an invalid sequence number.
3	FW_EXT_FW_DROP_L4_TYPE_INVALID_ACK_NUM	Occurs due to one of the following reasons: • When a packet’s ACK value is less than the connection’s oldest unacknowledged sequence number. • When a packet’s ACK value is greater than the connection’s next sequence number. • For SYN/ACK or ACK packets received during the three-way handshake, the sequence number is not equal to the initial sequence number plus 1.

Value	Event ID	Description
4	FW_EXT_FW_DROP_L4_TYPE_INVALID _TCP_INITIATOR	The first packet of a flow was not a SYN packet.
5	FW_EXT_FW_DROP_L4_TYPE_SYN _WITH_DATA	The SYN packet contains the payload and these SYN packet is not supported.
6	FW_EXT_FW_DROP_L4_TYPE_INVALI D_TCP_WIN_SCALE_OPTION	Invalid length for the TCP window-scale option.
7	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_SYNSENT_STATE	An invalid TCP segment was received in the SYNSENT state. Occurs due to one of the following reasons: • SYN/ACK has a payload. • SYN/ACK has other flags (push [PSH], urgent [URG], finish [FIN]) set. • Retransmit SYN message with a payload or invalid TCP flags (ACK, PSH, URG, FIN, RST) was received. • A non-SYN packet was received from the initiator.
8	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_SYNRCVD_STATE	A retransmitted SYN packet contains a payload or received a packet from the responder.
9	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_PKT_TOO_OLD	Packet is older (lesser than) than the receiver's current TCP window.
10	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_PKT_WIN_OVERFLOW	The sequence number of the packet is outside (greater than) the receiver's TCP window.
11	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_PYLD_AFTER_FIN_SEND	A packet containing a payload was received from the sender after a FIN message was received.
12	FW_EXT_FW_DROP_L4_TYPE_INVALID _FLAGS	TCP flags associated with the packet are not valid. This may occur for the following reasons: • Extra flags along with the SYN flag, are set in the initial packet. Only the SYN flag is allowed in the initial packet. • Expected SYN/ACK did not contain a SYN flag, or the SYN/ACK contained extraneous flags in the second packet of the three-way handshake.

Value	Event ID	Description
13	FW_EXT_FW_DROP_L4_TYPE_INVALID_SEQ	Invalid sequence number. Occurs due to one of the following reasons: • The sequence number is less than the ISN [9]. • The sequence number is equal to the ISN but not equal to a SYN packet. • If the receive window size is zero and the packet contains data, or if the sequence number is greater than the last ACK number. • Sequence number falls beyond the TCP window.
14	FW_EXT_FW_DROP_L4_TYPE_RETRANS _INVALID_FLAGS	A retransmitted packet was already acknowledged by the receiver.
15	FW_EXT_FW_DROP_L4_TYPE_L7_OOO _SEG	The packet contains a TCP segment that arrived prior to the expected next segment.
16	FW_EXT_FW_DROP_L4_TYPE_SYN _FLOOD_DROP	Maximum-incomplete sessions configured for the policy have been exceeded and the host is in block time.
17	FW_EXT_FW_DROP_L4_TYPE_MAX _HALFSESSION	Exceeded the number of allowed half-open sessions.
18	FW_EXT_FW_DROP_L4_TYPE_ TOO_MANY_PKTS	Exceeded the maximum number of simultaneous inspectable packets allowed per flow. The number is currently set to allow 25 simultaneous packets to be inspected. The simultaneous inspection prevents any one flow from monopolizing more than its share of processor resources.
19	FW_EXT_FW_DROP_L4_TYPE_TOO _MANY_ICMP_ERR_PKTS	Exceeded the maximum number of ICMP error packets allowed per flow. This log is triggered by the firewall base inspection.
20	FW_EXT_FW_DROP_L4_TYPE_UNEXPECT _TCP_PYLD	Retransmitted SYN/ACK from the responder included a payload. Payloads are not allowed during a TCP three-way handshake negotiation.
21	FW_EXT_FW_DROP_L4_TYPE_INTERNAL _ERR_UNDEFINED_DIR	Packet direction is undefined.

Value	Event ID	Description
22	FW_EXT_FW_DROP_L4_TYPE_SYN _IN_WIN	A TCP packet of an established session arrived with the SYN flag set. A SYN flag is not allowed after the initial two packets of the three-way handshake.
23	FW_EXT_FW_DROP_L4_TYPE_RST _IN_WIN	A TCP packet with the RST flag set was received with a sequence number that is outside the last received acknowledgment. The packet may be sent out of order.
24	FW_EXT_FW_DROP_L4_TYPE_ STRAY_SEG	An unexpected packet was received after the flow was torn down, or a packet was received from the responder before the initiator sent a valid SYN flag.
25	FW_EXT_FW_DROP_L4_TYPE_ RST_TO_RESP	A SYN/ACK flag was expected from the responder. However, a packet with an invalid sequence number was received. The zone-based firewall sent a RST flag to the responder.
26	FW_EXT_FW_DROP_L4_TYPE _I NTERNAL_ERR_ICMP_NO_NAT	The ICMP packet is NAT [10] translated; but internal NAT information is missing. An internal error.
27	FW_EXT_FW_DROP_L4_TYPE _ INTERNAL_ERR_ICMP_ALLOC_FAIL	Failed to allocate an ICMP error packet during an ICMP inspection.
28	FW_EXT_FW_DROP_L4_TYPE _INTERNAL_ERR_ICMP_GET_STAT_BLK_FAIL	The classification result did not have the required statistics memory. The policy information was not properly downloaded to the data plane.
29	FW_EXT_FW_DROP_L4_TYPE _INTERNAL_ERR_ICMP_DIR_NOT_IDENTIFIED	Packet direction is not defined.
30	FW_EXT_FW_DROP_L4_TYPE_ ICMP_SCB_CLOSE	Received an ICMP packet while the session is being torn down.
31	FW_EXT_FW_DROP_L4_TYPE_ ICMP_PKT_NO_IP_HDR	No IP header in the payload of the ICMP error packet.
32	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR_NO _IP_NO_ICMP	The ICMP error packet has no IP or ICMP, which is probably due to a malformed packet.
33	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR _PKTS_BURST	The ICMP error packet exceeded the burst limit of 10
34	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR _MULTIPLE_UNREACH	The ICMP error packet exceeded the “Unreachable” limit. Only the first unreachable packet is allowed to pass.

Value	Event ID	Description
35	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR _L4_INVALID_SEQ	The sequence number of the embedded packet does not match the sequence number of the TCP packet that triggers the ICMP error packet.
36	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR _L4_INVALID_ACK	The TCP packet contained in an ICMP error packet payload has an ACK flag that was not seen before.
37	FW_EXT_FW_DROP_L4_TYPE_ ICMP_PKT_TOO _SHORT	The ICMP error packet length is less than the IP header length plus the ICMP header length.
38	FW_EXT_FW_DROP_L4_TYPE_ SESSION_LIMIT	Resources exceeded the session limit while promoting for an imprecise channel.
39	FW_EXT_FW_DROP_L4_TYPE_ SCB_CLOSE	A TCP packet was received on a closed session.
40	FW_EXT_FW_DROP_INSP_TYPE_ POLICY_NOT_PRESENT	A policy is not present in a zone pair.
41	FW_EXT_FW_DROP_INSP_TYPE_ SESS_MISS_POLICY _NOT_PRESENT	A zone pair is configured in the same zone, but the zone does not have any policies.
44	FW_EXT_FW_DROP_INSP_TYPE_ CLASS_ACTION_DROP	The classification action is to drop the non-ICMP, TCP, and UDP packets.
45	FW_EXT_FW_DROP_INSP_TYPE_ PAM_LOOKUP_FAIL	The classification action is to drop the PAM entry.
48	FW_EXT_FW_DROP_INSP_TYPE_ INTERNAL_ERR_ GET_STAT_BLK_FAIL	Failed to get the statistic block from the classification result bytes.
49	FW_EXT_FW_DROP_SYNCOOKIE_ TYPE_SYNCOOKIE _MAX_DST	The maximum entry limit for SYN flood packets is reached.
50	FW_EXT_FW_DROP_SYNCOOKIE_ TYPE_INTERNAL _ERR_ALLOC_FAIL	Cannot allocate memory for the destination table entry.
51	FW_EXT_FW_DROP_SYNCOOKIE_ TYPE_SYN_COOKIE _TRIGGER	The SYN cookie logic is triggered. Indicates that the SYN/ACK with the SYN cookie was sent and the original SYN packet was dropped.
52	FW_EXT_FW_DROP_POLICY_ TYPE_FRAG_DROP	The first fragment of a VFR [11] packet is dropped and all associated remaining fragments will be dropped.
53	FW_EXT_FW_DROP_POLICY_ TYPE_ACTION_DROP	The classification action is to drop the packet.

Value	Event ID	Description
54	FW_EXT_FW_DROP_POLICY_ TYPE_ICMP_ACTION_DROP	The policy action of the ICMP embedded packet is DROP.
55	FW_EXT_FW_DROP_L7_TYPE_ NO_SEG	Layer 7 ALG [12] does not inspect inspect-segmented packets.
56	FW_EXT_FW_DROP_L7_TYPE_ NO_FRAG	Layer 7 ALG does not inspect fragmented packets.
57	FW_EXT_FW_DROP_L7_TYPE_ UNKNOWN_PROTO	Unknown application protocol type.
58	FW_EXT_FW_DROP_L7_TYPE_ ALG_RET_DROP	Layer 7 ALG inspection resulted in a packet drop.
59	FW_EXT_FW_DROP_NONSESSION _TYPE	Session creation has failed.
60	FW_EXT_FW_DROP_NO_NEW _SESSION_TYPE	During initial HA [13] states, a new session is not allowed.
61	FW_EXT_FW_DROP_NOT_ INITIATOR_TYPE	Not a session initiator packet.
62	FW_EXT_FW_DROP_INVALID _ZONE_TYPE	When default zones are not enabled, traffic is only allowed between interfaces that are associated with security zones.
64	FW_EXT_FW_DROP_NO_ FORWARDING_TYPE	The firewall is not configured.
65	FW_EXT_FW_DROP_ BACKPRESSURE_TYPE	The firewall backpressure can be enabled if HSL [14] is enabled, and the HSL logger was unable to send a log message. Backpressure will remain enabled until HSL is able to send a log.
66	FW_EXT_FW_DROP_L4_TYPE_INTERNAL _ERR _SYNFLOOD_ALLOC_HOSTDB_FAIL	During SYN processing, host rate limits are tracked. The host entry could not be allocated.
67	FW_EXT_FW_DROP_L4_TYPE_ SYNFLOOD_BLACKOUT_DROP	If the configured half-open connection limit is exceeded and blackout time is configured, all new connections to the specified IP address are dropped.
68	FW_EXT_FW_DROP_L7_TYPE_ PROMOTE_FAIL_NO_ZONE_PAIR	A failed policy. When an ALG attempts to promote a session because no zone pairs are configured, the policy fails.
69	FW_EXT_FW_DROP_L7_TYPE_ PROMOTE_FAIL_NO_POLICY	A failed policy. When an ALG attempts to promote a session due to no policy, the policy fails.

Value	Event ID	Description
	FW_EXT_FW_DROP_L4_TYPE_ONEFW _SCB_CLOSE	A packet is received after the Context-Aware firewall (CXSC) requested a teardown.
	FW_EXT_FW_DROP_L4_TYPE_ONEFW _FAIL_CLOSE	CXSC is not running.

[9] initial sequence number
[10] Network Address Translation
[11] virtual fragmentation and reassembly
[12] application layer gateway
[13] high availability
[14] high-speed logging

How to Configure Firewall High-Speed Logging

Enabling High-Speed Logging for Global Parameter Maps

By default, high-speed logging (HSL) is not enabled and firewall logs are sent to a logger buffer located in the Route Processor (RP) or the console. When HSL is enabled, logs are sent to an off-box, high-speed log collector. Parameter maps provide a means of performing actions on the traffic that reaches a firewall and a global parameter map applies to the entire firewall session table. Perform this task to enable high-speed logging for global parameter maps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **log dropped-packets**
5. **log flow-export v9 udp destination** *ip-address port-number*
6. **log flow-export template timeout-rate** *seconds*
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.

	Command or Action	Purpose
Step 3	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Configures a global parameter map and enters parameter-map type inspect configuration mode.
Step 4	**log dropped-packets** **Example:** `Device(config-profile)# log dropped-packets`	Enables dropped-packet logging.
Step 5	**log flow-export v9 udp destination** *ip-address port-number* **Example:** `Device(config-profile)# log flow-export v9 udp destination 10.0.2.0 5000`	Enables NetFlow event logging and provides the IP address and the port number of the log collector.
Step 6	**log flow-export template timeout-rate** *seconds* **Example:** `Device(config-profile) log flow-export template timeout-rate 5000`	Specifies the template timeout value.
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Enabling High-Speed Logging for Firewall Actions

Perform this task enable high-speed logging if you have configured inspect-type parameter maps. Parameter maps specify inspection behavior for the firewall and inspection parameter-maps for the firewall are configured as the inspect type.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **audit-trail on**
5. **alert on**
6. **one-minute** {**low** *number-of-connections* | **high** *number-of-connections*}
7. **tcp max-incomplete host** *threshold*
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** *class-map-name*
11. **inspect** *parameter-map-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect parameter-map-hsl`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** keyword, and enters parameter-map type inspect configuration mode.
Step 4	**audit-trail on** **Example:** `Device(config-profile)# audit-trail on`	Enables audit trail messages. • You can enable audit-trail to a parameter map to record the start, stop, and duration of a connection or session, and the source and destination IP addresses.
Step 5	**alert on** **Example:** `Device(config-profile)# alert on`	Enables stateful-packet inspection alert messages that are displayed on the console.
Step 6	**one-minute** {**low** *number-of-connections* | **high** *number-of-connections*} **Example:** `Device(config-profile)# one-minute high 10000`	Defines the number of new unestablished sessions that cause the system to start deleting half-open sessions and stop deleting half-open sessions.
Step 7	**tcp max-incomplete host** *threshold* **Example:** `Device(config-profile)# tcp max-incomplete host 100`	Specifies the threshold and blocking time values for TCP host-specific, denial of service (DoS) detection and prevention.
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 policy-map-hsl`	Creates an inspect-type policy map and enters policy map configuration mode.
Step 10	**class type inspect** *class-map-name* **Example:**	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.

	Command or Action	Purpose
	`Device(config-pmap)# class type inspect class-map-tcp`	
Step 11	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect parameter-map-hsl`	(Optional) Enables stateful packet inspection.
Step 12	**end** **Example:** `Device(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Configuration Examples for Firewall High-Speed Logging

Example: Enabling High-Speed Logging for Global Parameter Maps

The following example shows how to enable logging of dropped packets, and to log error messages in NetFlow Version 9 format to an external IP address:

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# log dropped-packets
Device(config-profile)# log flow-export v9 udp destination 10.0.2.0 5000
Device(config-profile)# log flow-export template timeout-rate 5000
Device(config-profile)# end
```

Example: Enabling High-Speed Logging for Firewall Actions

The following example shows how to configure high-speed logging (HSL) for inspect-type parameter-map parameter-map-hsl.

```
Device# configure terminal
Device(config)# parameter-map type inspect parameter-map-hsl
Device(config-profile)# audit trail on
Device(config-profile)# alert on
Device(config-profile)# one-minute high 10000
Device(config-profile)# tcp max-incomplete host 100
Device(config-profile)# exit
Device(config)# poliy-map type inspect policy-map-hsl
Device(config-pmap)# class type inspect class-map-tcp
Device(config-pmap-c)# inspect parameter-map-hsl
Device(config-pmap-c)# end
```

Additional References for Firewall High-Speed Logging

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 Firewall High-Speed Logging

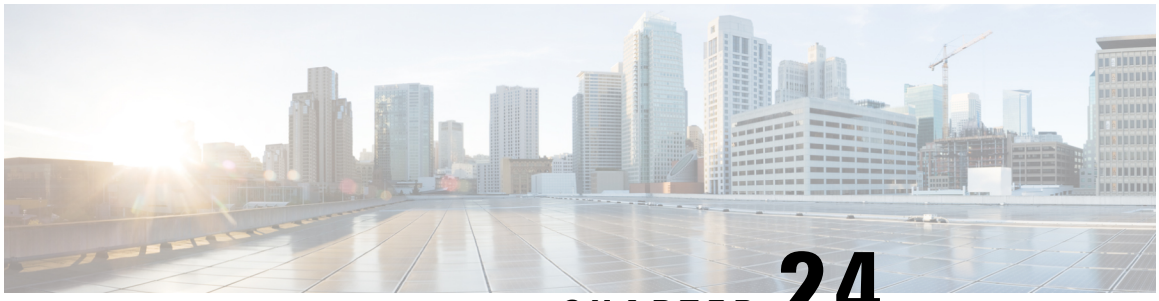
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 32: Feature Information for Firewall High-Speed Logging

Feature Name	Releases	Feature Information
Firewall High-Speed Logging	Cisco IOS XE Release 2.1	The Firewall High-Speed Logging Support feature introduces support for the firewall HSL using NetFlow Version 9 as the export format. The following commands were introduced or modified: **log dropped-packet**, **log flow-export v9 udp destination**, **log flow-export template timeout-rate**, **parameter-map type inspect global**.

Feature Name	Releases	Feature Information
Configuring Zone-based Firewall using High-Speed Logging	Cisco IOS XE Gibraltar 16.11.1	In this release, support was added for the source interface. The following commands were introduced or modified: **log flow-export v9 udp destination source interface** *interface-name*



CHAPTER 24

TCP Reset Segment Control

The TCP Reset Segment Control feature provides a mechanism to configure if a TCP reset (RST) segment should be sent when a session deletion occurs for half-close, half-open, or idle sessions.

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 TCP Reset Segment Control

TCP Reset Segment Control

The TCP header contains a flag known as the reset (RST) flag. A TCP segment is sent with the RST flag whenever a segment arrives that does not meet the criteria for a referenced connection. For example, a TCP segment is sent with a RST flag when a connection request is received on the destination port, but no process is listening at that port.

This behavior is defined in RFC 793, Transmission Control Protocol, for host-to-host communication and implemented by various vendors. However, for the network devices that reside on the network between hosts, specific rules have not been defined to determine if the device should send the TCP RST segment to the connection initiator, receiver, or both when sessions (half-open, idle, half-close) are cleared. Some devices send the TCP RST segment to both sender and receiver ports when a session is cleared, while some devices silently remove the session in the session table without sending out any TCP RST segments.

The TCP Reset Segment Control feature provides a mechanism to configure if a TCP RST segment should be sent when a session is cleared for half-close, half-open, or idle sessions.

A half-open session is an unestablished session initiated by a TCP synchronization (SYN) segment but is incomplete as only a TCP three-way handshake occurs and a timer is started.

TCP provides the ability for one end of a connection to terminate its output while still receiving data from the other end of the connection. This TCP state is called the half-close state. A session enters the half-close state when it receives the first TCP FIN segment and starts a timer. If another segment is received before the session timeout occurs, then the timer is restarted.

You can set the timeout value for half-open and half-close sessions by using the **tcp synwait-time** and **tcp finwait-time** commands respectively. The default timeout value is 30 seconds.

An idle session is a TCP session that is active between two devices and no data is transmitted by either of the devices for a prolonged period of time. You can set the timeout value for an idle session by using the **tcp idle-time** command. The default timeout value for idle sessions is 3600 seconds.

Once the timeout occurs on the TCP sessions and the session is cleared, the TCP RST segment is sent and the session will be reset only if the TCP reset segment control is configured on the sessions.

How to Configure TCP Reset Segment Control

Configuring TCP Reset for Half-Open Sessions

A half-open session is an unestablished session that is initiated by a TCP synchronization (SYN) segment but has an incomplete three-way handshake. A timer is started as soon as the incomplete three-way handshake occurs. You can set the timer values for a half-open session timeout by using the **tcp synwait-time** command. The default timeout value for these sessions is 30 seconds.

When the timeout occurs and the session is cleared on the half-open TCP session, the TCP reset (RST) segment is sent and the session will be reset only if the TCP reset segment control is configured on the sessions.

If you configure the **tcp half-open reset on** command, the TCP RST segment is sent to both ends of the half-open session when the session is cleared. If you configure the **tcp half-open reset off** command, the TCP RST segment is not transmitted when the session is cleared.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **tcp synwait-time** *seconds*
5. **tcp half-open reset** {**off** | **on**}
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** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap-name`	(Optional) Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** keyword and enters parameter-map type inspect configuration mode.
Step 4	**tcp synwait-time** *seconds* **Example:** `Device(config-profile)# tcp synwait-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session.
Step 5	**tcp half-open reset** {**off** | **on**} **Example:** `Device(config-profile)# tcp half-open reset on`	Specifies whether the TCP RST segment should be sent when timeout occurs and the session is cleared for a half-open session.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuring TCP Reset for Half-Close Sessions

TCP provides the ability for one end of a connection to terminate its output, while still receiving data from the other end of the connection. This TCP state is called the half-close state. A session enters the half-close state when it receives the first TCP finish (FIN) segment and starts a timer. If another segment is received before the session timeout occurs, then the timer is restarted. You can set the timeout value for a half-close session by using the **tcp finwait-time** command. The default timeout value for half-close sessions is 30 seconds.

Once the timeout occurs on the half-close TCP session, the TCP RST segment is sent and the session will be reset only if the TCP reset segment control is configured on the sessions.

If you configure the **tcp half-close reset on** command, the TCP RST segment is sent to both ends of the half-open session when timeout occurs and the session is cleared. If you configure the **tcp half-close reset off** command, the TCP RST segment is not transmitted when the session timeout occurs and the session is cleared.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **tcp finwait-time** *seconds*
5. **tcp half-close reset** {**off** | **on**}
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** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap-name`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** keyword and enters parameter-map type inspect configuration mode.
Step 4	**tcp finwait-time** *seconds* **Example:** `Device(config-profile)# tcp finwait-time 10`	(Optional) Specifies how long a TCP session will be managed after the firewall detects a FIN-exchange.
Step 5	**tcp half-close reset** {**off** | **on**} **Example:** `Device(config-profile)# tcp half-close reset on`	Specifies whether the TCP RST segment should be sent when session deletion occurs on a half-open session.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuring TCP Reset for Idle Sessions

An idle session is a TCP session that is active between two devices and no data is transmitted by either device for a prolonged period of time. You can set the timeout value for an idle session by using the **tcp idle-time** command. The default timeout value for idle sessions is 3600 seconds.

Once the timeout occurs on the idle TCP session, the TCP RST segment is sent and the session will be reset if the TCP reset segment control is configured on the session.

If you configure the **tcp idle reset on** command, the TCP RST segment is sent to both ends of the idle session when timeout occurs and the session is cleared. If you configure the **tcp idle reset off** command, the TCP RST segment is not transmitted when the session timeout occurs and the session is cleared.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **tcp idle-time** *seconds*
5. **tcp idle reset** {**off** | **on**}
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** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap-name`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** keyword and enters parameter-map type inspect configuration mode.
Step 4	**tcp idle-time** *seconds* **Example:** `Device(config-profile)# tcp idle-time 90`	(Optional) Configures the timeout for TCP sessions.
Step 5	**tcp idle reset** {**off** | **on**} **Example:** `Device(config-profile)# tcp idle reset on`	Specifies whether the TCP RST segment should be sent when session deletion occurs on an idle session.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuration Examples for TCP Reset Segment Control

Example: Configuring TCP Reset for Half-Open Sessions

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect pmap-name
Device(config-profile)# tcp synwait-time 10
```

```
Device(config-profile)# tcp half-open reset on
Device(config-profile)# end
```

Example: Configuring TCP Reset for Half-Close Sessions

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect pmap-name
Device(config-profile)# tcp finwait-time 10
Device(config-profile)# tcp half-close reset on
Device(config-profile)# end
```

Example: Configuring TCP Reset for Idle Sessions

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect pmap-name
Device(config-profile)# tcp idle-time 90
Device(config-profile)# tcp idle reset on
Device(config-profile)# end
```

Additional References for TCP Reset Segment Control

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

Standards and RFCs

Standard/RFC	Title
RFC 793	Transmission Control 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

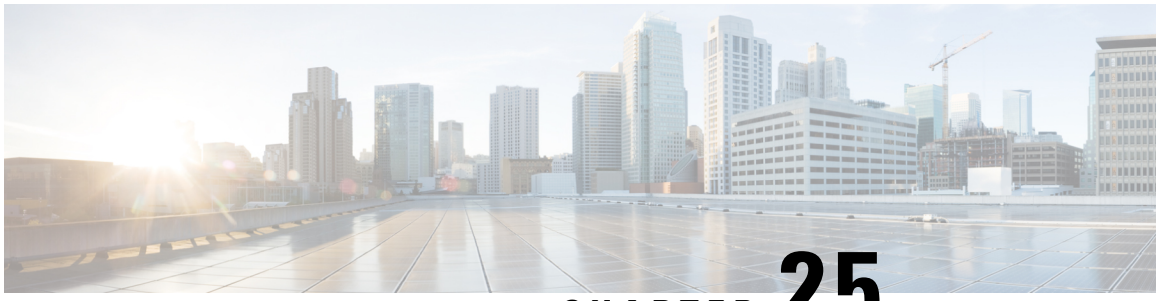
Feature Information for TCP Reset Segment Control

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 33: Feature Information for TCP Reset Segment Control

Feature Name	Releases	Feature Information
TCP Reset Segment Control	Cisco IOS XE Release 3.8S	The TCP Reset Segment Control feature provides a consistent mechanism to configure if the TCP RST bits should be sent out when a session is cleared for half-open, half-close, and idle sessions. The following commands were introduced or modified: **tcp idle reset**, **tcp half-close reset**, and **tcp half-open reset**.



CHAPTER 25

Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

The Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall feature disables the strict checking of the TCP window-scaling option in a 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.

Information About Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

Loose Checking Option for TCP Window Scaling Overview

TCP provides various TCP extensions to improve performance over high-bandwidth and high-speed data paths. One such extension is the TCP window-scaling option. The loose-checking option for TCP window-scaling turns off strict checking of the window-scaling option described in RFC 1323.

A larger window size is recommended to improve TCP performance in network paths with large bandwidth-delay product characteristics that are called Long Fat Networks (LFNs). TCP window scaling expands the definition of the TCP window to 32 bits and then uses a scale factor to carry this 32-bit value in the 16-bit window field of the TCP header. The window size can increase to a scale factor of 14. Typical applications use a scale factor of 3 when deployed in LFNs.

A firewall implementation enforces strict checking of the TCP window-scaling option. A firewall drops SYN/ACK packets that have the TCP window-scaling option if it was not offered in the initial synchronization (SYN) packet for the TCP three-way handshake. The window-scale option is sent only in a SYN segment, which is a segment with the SYN bit on. Therefore, the window scale is fixed in each direction when a connection is opened.

Use the **tcp window-scale-enforcement loose** command to disable the strict checking of the TCP window-scaling option in TCP SYN segments.

How to Configure Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

Configuring the TCP Window-Scaling Option for a Firewall

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** {*parameter-map-name* | **global** | **default**}
4. **tcp window-scale-enforcement loose**
5. **exit**
6. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
7. **match protocol** [*parameter-map*] [**signature**]
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** *class-map-name*
11. **inspect** [*parameter-map-name*]
12. **exit**
13. **class** *name*
14. **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	**parameter-map type inspect** {*parameter-map-name* | **global** | **default**} **Example:** `Device(config)# parameter-map type inspect pmap-fw`	Configures an inspect parameter map and enters profile configuration mode.
Step 4	**tcp window-scale-enforcement loose** **Example:** `Device(config-profile)# tcp window-scale-enforcement loose`	Disables the strict checking of the TCP window-scaling option in a firewall.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits profile configuration mode and returns to global configuration mode.
Step 6	**class-map type inspect** {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any internet-traffic-class`	Creates an inspect-type class map and enters QoS class-map configuration mode.
Step 7	**match protocol** [*parameter-map*] [**signature**] **Example:** `Device(config-cmap)# match protocol tcp`	Configures a match criteria for a class map on the basis of the specified protocol.
Step 8	**exit** **Example:** `Device(config-cmap)# exit`	Exits the QoS 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 private-internet-policy`	Creates an 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 internet-traffic-class`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 11	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect pmap-fw`	Enables stateful packet inspection.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and returns to QoS policy-map configuration mode.
Step 13	**class** *name* **Example:** `Device(config-pmap)# class class-default`	Associates the map class with a specified data-link connection identifier (DLCI).
Step 14	**end** **Example:** `Device(config-pmap)# end`	Exits QoS policy-map configuration mode and returns to privileged EXEC mode.

Configuring a Zone and Zone Pair for a TCP Window Scaling

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address*
5. **zone-member security** *security-zone-name*
6. **exit**
7. **interface** *type number*
8. **ip address** *ip-address*
9. **zone-member security** *security-zone-name*
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 0/1/5`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip address** *ip-address* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Assigns an interface IP address.
Step 5	**zone-member security** *security-zone-name* **Example:** `Device(config-if)# zone-member security private`	Configures the interface as a zone member.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/6`	Specifies an interface and enters interface configuration mode.
Step 8	**ip address** *ip-address* **Example:** `Device(config-if)# ip address 209.165.200.225 255.255.255.0`	Assigns an IP address to an interface.
Step 9	**zone-member security** *security-zone-name* **Example:** `Device(config-if)# zone-member security internet`	Configures an interface as a zone member.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for TCP Window-Scaling

Example: Configuring the TCP Window-Scaling Option for a Firewall

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect pmap-fw
Device(config-profile)# tcp window-scale-enforcement loose
Device(config-profile)# exit
Device(config)# class-map type inspect match-any internet-traffic-class
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect private-internet-policy
Device(config-pmap)# class type inspect internet-traffic-class
Device(config-pmap-c)# inspect pmap-fw
```

```
Device(config-pmap-c)#exit
Device(config-pmap)# class class-default
Device(config-pmap)#end
```

Example: Configuring a Zone and Zone Pair for TCP Window Scaling

```
Device# enable
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/5
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# zone-member security private
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/1/6
Device(config-if)# ip address 209.165.200.225 255.255.255.0
Device(config-if)# zone-member security internet
Device(config-if)# end
```

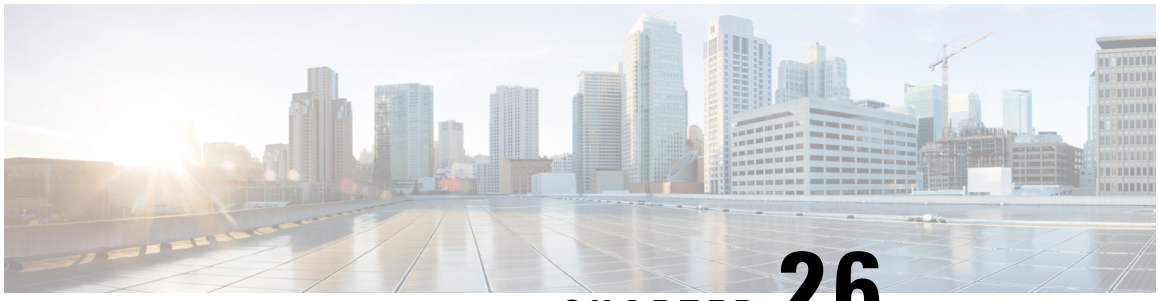
Feature Information for Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

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 34: Feature Information for Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

Feature Name	Releases	Feature Information
Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall	Cisco IOS XE Release 3.10S	Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall feature disables the strict checking of the TCP Window Scaling option in an IOS-XE firewall. The following command was introduced or modified: **tcp window-scale-enforcement loose**. In Cisco IOS XE Release 3.10S, support was added for the Cisco CSR 1000V Series Routers.



CHAPTER 26

Enabling ALGs and AICs in Zone-Based Policy Firewalls

Zone-based policy firewalls support Layer 7 application protocol inspection along with application-level gateways (ALGs) and application inspection and control (AIC). Layer 7 application protocol inspection helps to verify the protocol behavior and identify unwanted or malicious traffic that passes through a security module.

Prior to the introduction of Enabling ALGs and AICs in Zone-Based Policy Firewalls feature, the Layer 7 protocol inspection was automatically enabled along with the ALG/AIC configuration. With this feature you can enable or disable Layer 7 inspection by using the **no application-inspect** command.

This module provides an overview of the Enabling ALGs and AICs in Zone-Based Policy Firewalls feature and describes how to configure it.

Finding Feature Information

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

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

Information About Enabling ALGs and AICs in Zone-Based Policy Firewalls

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

Enabling Layer 7 Application Protocol Inspection Overview

Zone-based policy firewalls support Layer 7 protocol inspection along with application-level gateways (ALG) and application inspection and control (AIC). Layer 7 protocol inspection is automatically enabled along with the ALG/AIC configuration.

Layer 7 application protocol inspection is a technique that interprets or understands application-layer protocols and performs appropriate firewall or Network Address Translation (NAT) action. Certain applications require special handling of the data portion of a packet when the packet passes through the security module on a device. Layer 7 application protocol inspection helps to verify the protocol behavior and identify unwanted or malicious traffic that passes through the security module. Based on the configured traffic policy, the security module accepts or rejects packets to ensure the secure use of applications and services.

Sometimes, application inspection implementation issues can cause application packet drop and make networks unstable. Prior to the introduction of the Enabling ALGs and AICs in Zone-Based Policy Firewall feature, to disable application inspection you had to define an access control list (ACL) with the target Layer 7 protocol port define a class map that matches this ACL and matches either the TCP or UDP protocol to bypass the inspection for a specific Layer 7 protocol.

With the introduction of the Enabling ALGs and AICs in Zone-Based Policy Firewall feature, you can enable or disable Layer 7 protocol inspection for a specific protocol or for all supported Layer 7 protocols with the **application-inspect** command. Any configuration changes to a parameter map applies only to new sessions. For example, when you disable FTP Layer 7 inspection, the newly created sessions skip FTP Layer 7 inspection, while existing sessions before the configuration change will perform FTP Layer 7 inspection. For all sessions to perform the configuration change, you must delete all sessions and re-create them.

You can enable Layer 7 application protocol inspection for an individual parameter map or for a global firewall.

How to Enable ALGs and AICs in Zone-Based Policy Firewalls

Enabling Layer 7 Application Protocol Inspection on Firewalls

Application protocol inspection is enabled by default. Use the **no application-inspect** command to disable application protocol inspection.

Use the **application-inspect** command to reconfigure application protocol inspection, if you have disabled it for any reason. Configure either the **parameter-map type inspect** command or the **parameter-map type inspect-global** command before configuring the **application-inspect** command.

You can only configure either the **parameter-map type inspect** command or the **parameter-map type inspect-global** command at any time.

Use the

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **parameter-map type inspect** *parameter-map-name*
   - **parameter-map type inspect-global**
4. **application-inspect** {**all** | *protocol-name*}
5. **exit**
6. **class-map type inspect** {**match-all** | **match-any**} *class-map-name*
7. **match protocol** *protocol-name*
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** {*class-map-name* | **class-default**}
11. **inspect** *parameter-map-name*
12. **exit**
13. **class** {*class-map-name* | **class-default**}
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 terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 3	Do one of the following:
	• **parameter-map type inspect** *parameter-map-name*
	• **parameter-map type inspect-global**
	Example:
	`Device(config)# parameter-map type inspect pmap-fw`
	or
	`Device(config)# parameter-map type inspect-global`
		• (Optional) Enables an inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
		• (Optional) Enables a global parameter map and enters parameter-map type inspect configuration mode.

Step 4	**application-inspect** {**all** | *protocol-name*} **Example:** `Device(config-profile)# application-inspect msrpc`	Enables application inspection for the specified protocols.
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-all** | **match-any**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any internet-traffic-class`	Creates an inspect type class map and enters class map configuration mode.
Step 7	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol msrpc`	Configures a match criterion for a class map based on the 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 private-internet-policy`	Creates an inspect type policy map and enters policy map configuration mode.
Step 10	**class type inspect** {*class-map-name* | **class-default**} **Example:** `Device(config-pmap)# class type inspect internet-traffic-class`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 11	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap-fw`	Enables 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-map-name* | **class-default**} **Example:** `Device(config-pmap)# class class-default`	Specifies the default class so that you can configure or modify the policy.
Step 14	**end** **Example:** `Device(config-pmap)# end`	Exits policy map configuration mode and returns to privileged EXEC mode.

Configuring Zones for Enabling Layer 7 Application Protocol Inspection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {**default** | *security-zone*}
4. **exit**
5. **zone security** {**default** | *security-zone*}
6. **exit**
7. **zone-pair security** *zone-pair* **source** *source-zone* **destination** *destination-zone*
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *security-zone*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *security-zone*
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.

	Command or Action	Purpose
Step 3	**zone security** {**default** | *security-zone*} **Example:** `Device(config)# zone security private`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • You need two security zones to create a zone pair: a source and a destination zone.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** {**default** | *security-zone*} **Example:** `Device(config)# zone security internet`	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-pair security** *zone-pair* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security private-internet source private destination internet`	Creates a zone pair and enters security zone pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect private-internet-policy`	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.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *security-zone* **Example:** `Device(config-if)# zone-member security private`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into 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 apply a policy. If the policy permits traffic, traffic can flow through that interface.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2/2`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *security-zone* **Example:** `Device(config-if)# zone-member security internet`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Enabling ALGs and AICs in Zone-Based Policy Firewalls

Example: Enabling Layer 7 Application Protocol Inspection on Firewalls

The following example shows how to enable Layer 7 application protocol inspection after configuring the **parameter-map type inspect** command. You can enable application inspection after configuring the **parameter-map type inspect-global** command also.

You can only configure either the **parameter-map type inspect** or the **parameter-map type inspect-global** command at any time.

```
Device# configure terminal
Device(config)# parameter-map type inspect pmap-fw
Device(config-profile)# application-inspect msrpc
Device(config-profile)# exit
Device(config)# class-map type inspect match-any internet-traffic-class
Device(config-cmap)# match protocol msrpc
Device(config-cmap)# exit
Device(config)# policy-map type inspect private-internet-policy
Device(config-pmap)# class type inspect internet-traffic-class
Device(config-pmap-c)# inspect pmap-fw
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap)# end
```

Example: Configuring Zones for Enabling Layer 7 Application Protocol Inspection

```
Device# configure terminal
Device(config)# zone security private
Device(config-sec-zone)# exit
Device(config)# zone security internet
Device(config-sec-zone)# exit
Device(config)# zone-pair security private-internet source private destination internet
Device(config-sec-zone-pair)# service-policy type inspect private-internet-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# zone-member security private
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/2/2
Device(config-if)# zone-member security internet
Device(config-if)# end
```

Additional References for Enabling ALGs and AICs in Zone-Based Policy Firewalls

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

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

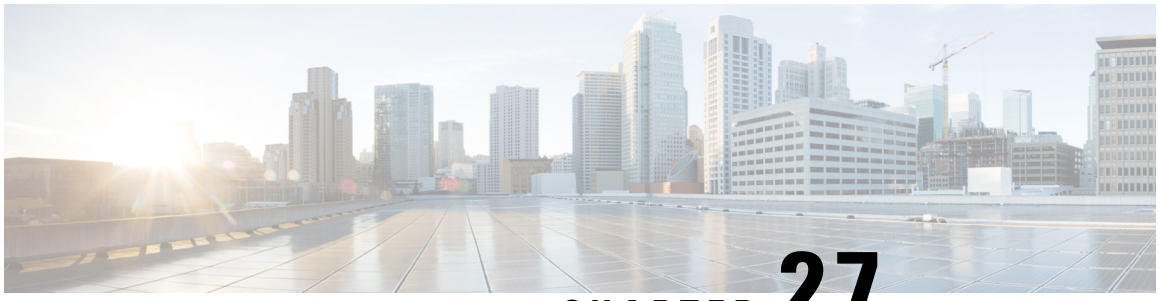
Feature Information for Enabling ALGs and AICs in Zone-Based Policy Firewalls

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 35: Feature Information for Enabling ALGs and AICs in Zone-Based Policy Firewalls

Feature Name	Releases	Feature Information
Enabling ALGs and AICs in Zone-Based Policy Firewalls	Cisco IOS XE Release 3.11S	Zone-based policy firewalls support Layer 7 application protocol inspection along with application-level gateways (ALGs) and application inspection and control (AIC). Layer 7 application protocol inspection helps to verify the protocol behavior and identify unwanted or malicious traffic that passes through security module. Prior to the introduction of Enabling ALGs and AICs in Zone-Based Policy Firewalls feature, the Layer 7protocol inspection was automatically enabled along with the ALG/AIC configuration. With this feature you can enable or disable Layer 7 inspection by using the no application-inspect command. In Cisco IOS XE Release 3.11S, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers, Cisco 4400 Series Integrated Services Routers, and Cisco Cloud Services Routers 1000V. The following commands were introduced or modified: **application-inspect**, **show parameter-map type inspect**, and **show platform software firewall**.



CHAPTER 27

Configuring Firewall TCP SYN Cookie

The Firewall TCP SYN Cookie feature protects your firewall from TCP SYN-flooding attacks. TCP SYN-flooding attacks are a type of denial-of-service (DoS) attack. Usually, TCP synchronization (SYN) packets are sent to a targeted end host or a range of subnet addresses behind the firewall. These TCP SYN packets have spoofed source IP addresses. A spoofing attack is when a person or a program pretends to be another by falsifying data and thereby gaining an illegitimate advantage. TCP SYN-flooding can take up all resources on a firewall or an end host, thereby causing DoS to legitimate traffic. To prevent TCP SYN-flooding on a firewall and the end hosts behind the firewall, you must configure the Firewall TCP SYN Cookie 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 Configuring Firewall TCP SYN Cookie

- Because a default zone does not support zone type parameter map, you cannot configure the Firewall TCP SYN Cookie feature for a default zone.
- The Firewall TCP SYN Cookie feature does not support per-subscriber firewall.

Information About Configuring Firewall TCP SYN Cookie

TCP SYN Flood Attacks

The Firewall TCP SYN Cookie feature implements software to protect the firewall from TCP SYN-flooding attacks, which are a type of DoS attack.

A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Because these messages have unreachable return addresses, the connections cannot be established. The resulting volume of unresolved open connections eventually overwhelms the server and can cause it to deny service to valid requests, thereby preventing legitimate users from connecting to a website, accessing e-mail, using FTP service, and so on.

SYN flood attacks are divided into two types:

- Host flood—SYN flood packets are sent to a single host aiming to utilize all resources on that host.
- Firewall session table flood—SYN flood packets are sent to a range of addresses behind the firewall, with the aim of exhausting the session table resources on the firewall and thereby denying resources to the legitimate traffic going through the firewall.

The Firewall TCP SYN Cookie feature helps prevent SYN-flooding attacks by intercepting and validating TCP connection requests. The firewall intercepts TCP SYN packets that are sent from clients to servers. When the TCP SYN cookie is triggered, it acts on all SYN packets that are destined to the configured VPN Routing and Forwarding (VRF) or zone. The TCP SYN cookie establishes a connection with the client on behalf of the destination server and another connection with the server on behalf of the client and knits together the two half-connections transparently. Thus, connection attempts from unreachable hosts will never reach the server. The TCP SYN cookie intercepts and forwards packets throughout the duration of the connection.

The Firewall TCP SYN Cookie feature provides session table SYN flood protection for the global routing domain and for the VRF domain. Because the firewall saves sessions in a global table, you can configure a limit to the number of TCP half-opened sessions. A TCP half-opened session is a session that has not reached the established state. In a VRF-aware firewall, you can configure a limit to the number of TCP half-opened sessions for each VRF. At both the global level and at the VRF level, when the configured limit is reached, the TCP SYN cookie verifies the source of the half-opened sessions before creating more sessions.

How to Configure Firewall TCP SYN Cookie

Configuring Firewall Host Protection

TCP SYN packets are sent to a single host with the aim of taking over all resources on the host. You can configure host protection only for the source zone. Configuring protection on the destination zone will not protect the destination zone from TCP SYN attacks.

Perform this task to configure the firewall host protection.

Note You can specify the **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-zone** *zone-pmap-name*
4. **tcp syn-flood rate per-destination** *maximum-rate*
5. **max-destination** *limit*
6. **exit**
7. **zone security** *zone-name*
8. **protection** *parameter-map-name*
9. **exit**
10. **show parameter-map type inspect-zone** *zone-pmap-name*
11. **show zone security**
12. **show policy-firewall stats zone** *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-zone** *zone-pmap-name* **Example:** `Router(config)# parameter-map type inspect-zone zone-pmap`	Configures an inspect zone type parameter map and enters profile configuration mode.
Step 4	**tcp syn-flood rate per-destination** *maximum-rate* **Example:** `Router(config-profile)# tcp syn-flood rate per-destination 400`	Configures the number of SYN flood packets per second for each destination address. • If the rate of SYN packets sent to a particular destination address exceeds the per-destination limit, the firewall starts processing SYN cookies for SYN packets that are routed to the destination address.
Step 5	**max-destination** *limit* **Example:** `Router(config-profile)# max-destination 10000`	Configures the maximum number of destinations that the firewall can track for a zone. • The firewall drops the SYN packets if the maximum destination crosses the limit that is configured by using the *limit* argument.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-profile)# exit`	Exits profile configuration mode and enters global configuration mode.
Step 7	**zone security** *zone-name* **Example:** `Router(config)# zone security secure-zone`	Configures a security zone and enters security zone configuration mode.
Step 8	**protection** *parameter-map-name* **Example:** `Router(config-sec-zone)# protection zone-pmap`	Configures protection for the specified zone using the parameter map.
Step 9	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration and enters privileged EXEC mode.
Step 10	**show parameter-map type inspect-zone** *zone-pmap-name* **Example:** `Router# show parameter-map type inspect-zone zone-pmap`	(Optional) Displays details about the inspect zone type parameter map.
Step 11	**show zone security** **Example:** `Router# show zone security`	(Optional) Displays zone security information.
Step 12	**show policy-firewall stats zone** *zone-name* **Example:** `Router# show policy-firewall stats zone secure-zone`	(Optional) Displays how many SYN packets exceeded the packet limit and were processed by SYN cookies.

Configuring Firewall Session Table Protection

TCP SYN packets are sent to a range of addresses behind the firewall aiming to exhaust the session table resources on the firewall, thereby denying resources to the legitimate traffic going through the firewall. You can configure firewall session table protection either for the global routing domain or for the VRF domain.

Configuring Firewall Session Table Protection for Global Routing Domain

Perform this task to configure firewall session table protection for global routing domains.

Note A global parameter map takes effect on the global routing domain and not at the router level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **tcp syn-flood limit** *number*
5. **end**
6. **show policy-firewall stats vrf global**

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 global** **Example:** `Router(config)# parameter-map type inspect global`	Configures a global parameter map and enters profile configuration mode.
Step 4	**tcp syn-flood limit** *number* **Example:** `Router(config-profile)# tcp syn-flood limit 500`	Limits the number of TCP half-open sessions that triggers SYN cookie processing for new SYN packets.
Step 5	**end** **Example:** `Router(config-profile)# end`	Exits profile configuration mode and enters privileged EXEC mode.
Step 6	**show policy-firewall stats vrf global** **Example:** `Router# show policy-firewall stats vrf global`	(Optional) Displays the status of the global VRF firewall policy. • The command output also displays how many TCP half-open sessions are present.

Configuring Firewall Session Table Protection for VRF Domain

Perform this task to configure the firewall session table protection for VRF domains.

Note You can specify the **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-pmap-name*
4. **tcp syn-flood limit** *number*
5. **exit**
6. **parameter-map type inspect global**
7. **vrf** *vrf-name* **inspect** *parameter-map-name*
8. **end**
9. **show parameter-map type inspect-vrf**
10. **show policy-firewall stats vrf** *vrf-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-vrf** *vrf-pmap-name* **Example:** `Router(config)# parameter-map type inspect-vrf vrf-pmap`	Configures an inspect-VRF type parameter map and enters profile configuration mode.
Step 4	**tcp syn-flood limit** *number* **Example:** `Router(config-profile)# tcp syn-flood limit 200`	Limits the number of TCP half-open sessions that triggers SYN cookie processing for new SYN packets.
Step 5	**exit** **Example:**	Exits profile configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Router(config-profile)# exit`	
Step 6	**parameter-map type inspect global** **Example:** `Router(config)# parameter-map type inspect global`	Binds the inspect-VRF type parameter map to a VRF and enters profile configuration mode.
Step 7	**vrf** *vrf-name* **inspect** *parameter-map-name* **Example:** `Router(config-profile)# vrf vrf1 inspect vrf-pmap`	Binds the parameter map to the VRF.
Step 8	**end** **Example:** `Router(config-profile)# end`	Exits profile configuration mode and enters privileged EXEC mode.
Step 9	**show parameter-map type inspect-vrf** **Example:** `Router# show parameter-map type inspect-vrf`	(Optional) Displays information about inspect VRF type parameter map.
Step 10	**show policy-firewall stats vrf** *vrf-name* **Example:** `Router# show policy-firewall stats vrf vrf-pmap`	(Optional) Displays the status of the VRF firewall policy. • The command output also displays how many TCP half-open sessions are present.

Configuration Examples for Firewall TCP SYN Cookie

Example Configuring Firewall Host Protection

The following example shows how to configure the firewall host protection:

```
Router(config)# parameter-map type inspect-zone zone-pmap
Router(config-profile)# tcp syn-flood rate per-destination 400
Router(config-profile)# max-destination 10000
Router(config-profile)# exit
Router(config)# zone security secure-zone
```

```
Router(config-sec-zone)# protection zone-pmap
```

Example Configuring Firewall Session Table Protection

Global Parameter Map

The following example shows how to configure firewall session table protection for global routing domains:

```
Router# configure terminal
Router(config)# parameter-map type inspect global
Router(config-profile)# tcp syn-flood limit 500
Router(config-profile)# end
```

Inspect-VRF Type Parameter Map

The following example shows how to configure firewall session table protection for VRF domains:

```
Router# configure terminal
Router(config)# parameter-map type inspect-vrf vrf-pmap
Router(config-profile)# tcp syn-flood limit 200
Router(config-profile)# exit
Router(config)# parameter-map type inspect global
Router(config-profile)# vrf vrf1 inspect vrf-pmap
Router(config-profile)# end
```

Additional References for Firewall TCP SYN Cookie

Related Documents

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

Related Topic	Document Title
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

Technical Assistance

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

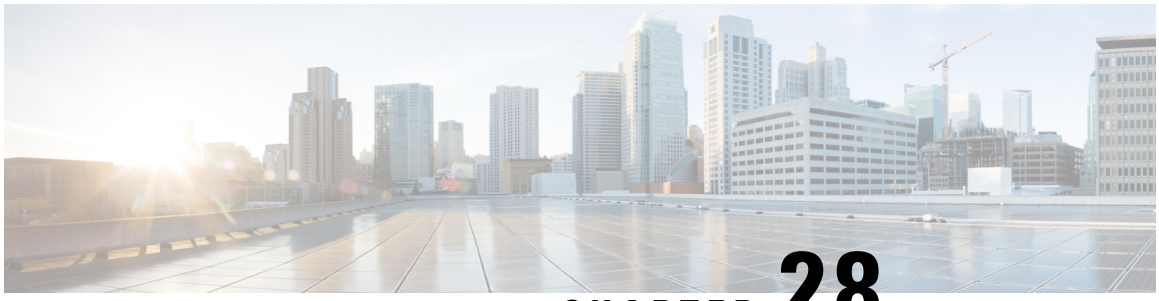
Feature Information for Configuring Firewall TCP SYN Cookie

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 36: Feature Information for Configuring Firewall TCP SYN Cookie

Feature Name	Releases	Feature Information
Firewall TCP SYN Cookie	Cisco IOS XE Release 3.3S	The Firewall TCP SYN Cookie feature protects your firewall from TCP SYN-flooding attacks. TCP SYN-flooding attacks are a type of DoS attack. Usually, TCP SYN packets are sent to a targeted end host or a range of subnet addresses behind the firewall. These TCP SYN packets have spoofed source IP addresses. A spoofing attack is when a person or a program pretends to be another by falsifying data and thereby gaining an illegitimate advantage. The TCP SYN-flooding can take up all the resource on a firewall or an end host, thereby causing DoS to legitimate traffic. To prevent TCP SYN-flooding on a firewall and the end hosts behind the firewall, you must configure the Firewall TCP SYN Cookie feature. The following commands were introduced or modified: **parameter-map type inspect-vrf**, **parameter-map type inspect-zone**, **parameter-map type inspect global**, **show policy-firewall stats**, **tcp syn-flood rate per-destination**, **tcp syn-flood limit**.



CHAPTER 28

Object Groups for ACLs

The Object Groups for ACLs feature lets you classify users, devices, or protocols into groups and apply these groups to access control lists (ACLs) to create access control policies for these groups. This feature lets you use object groups instead of individual IP addresses, protocols, and ports, which are used in conventional ACLs. This feature allows multiple access control entries (ACEs). You can use each ACE to allow an entire group of users to access a group of servers or services or to deny them access; thereby reducing the size of an ACL and improving manageability.

This module describes object-group ACLs with zone-based policy firewalls and how to configure them 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.

Restrictions for Object Groups for ACLs

The following restrictions apply to the Object Groups for ACLs feature on zone-based firewalls:

- IPv6 is not supported.
- Dynamic and per-user access control lists (ACLs) are not supported.
- You cannot remove an object group or make an object group empty if it is used in an ACL.

- ACL statements using object groups will be ignored on packets that are sent to RP for processing.
- Object groups are supported only for IP extended ACLs.

Information About Object Groups for ACLs

Overview of Object Groups for ACLs

In large networks, the number of lines in an access control list (ACL) can be large (hundreds of lines) and difficult to configure and manage, especially if the ACLs frequently change. Object group-based ACLs are smaller, more readable, and easier to configure and manage. Object-group-based ACLs simplify static ACL deployments for large user access environments on Cisco IOS routers. The zone-based firewall benefits from object groups, because object groups simplify policy creation (for example, group A has access to group A services).

You can configure conventional access control entries (ACEs) and ACEs that refer to object groups in the same ACL. You can use object-group-based ACLs with quality of service (QoS) match criteria, zone-based policy firewall, Dynamic Host Configuration Protocol (DHCP), and any other features that use extended ACLs.

In addition, you can use object-group-based ACLs with multicast traffic. When there are many inbound and outbound packets, using object group-based ACLs increases performance compared to conventional ACLs. Also, in large configurations, this feature reduces the storage required in NVRAM, because you need not define an individual ACE for every address and protocol pairing.

Integration of Zone-Based Firewalls with Object Groups

Zone-based firewalls use object-group access control lists (ACLs) to apply policies to specific traffic. You define an object-group ACL, associate it with a zone-based firewall policy, and apply the policy to a zone pair to inspect the traffic.

In Cisco IOS XE Release 3.12S, only expanded object-group ACLs are supported with firewalls.

The following features work with object groups that are configured on a firewall:

- Static and dynamic network address translation (NAT)
- Service NAT (NAT that supports non-standard FTP port numbers configured by the **ip nat service** command)
- FTP application layer gateway (ALG)
- Session Initiation Protocol (SIP) ALG

In a class map, you can configure a maximum of 64 matching statements using the **match access-group** command.

Objects Allowed in Network Object Groups

A network object group is a group of any of the following objects:

- Any IP address—includes a range from 0.0.0.0 to 255.255.255.255 (This is specified using the **any** command.)

- Host IP addresses
- Hostnames
- Other network object groups
- Subnets

- Host IP addresses
- Network address of group members
- Nested object groups

Objects Allowed in Service Object Groups

A service object group is a group of any of the following objects:

- Source and destination protocol ports (such as Telnet or Simple Network Management Protocol [SNMP])
- Internet Control Message Protocol (ICMP) types (such as echo, echo-reply, or host-unreachable)
- Top-level protocols (such as Encapsulating Security Payload [ESP], TCP, or UDP)
- Other service object groups

ACLs Based on Object Groups

All features that use or reference conventional access control lists (ACLs) are compatible with object-group-based ACLs, and the feature interactions for conventional ACLs are the same with object-group-based ACLs. This feature extends the conventional ACLs to support object-group-based ACLs and also adds new keywords and the source and destination addresses and ports.

You can add, delete, or change objects in an object group membership list dynamically (without deleting and redefining the object group). Also, you can add, delete, or change objects in an object group membership list without redefining the ACL access control entry (ACE) that uses the object group. You can add objects to groups, delete them from groups, and then ensure that changes are correctly functioning within the object-group-based ACL without reapplying the ACL to the interface.

You can configure an object-group-based ACL multiple times with a source group only, a destination group only, or both source and destination groups.

You cannot delete an object group that is used within an ACL or a class-based policy language (CPL) policy.

Guidelines for Object Group ACLs

- Object groups must have unique names. For example, to create a network object group named “Engineering” and a service object group named “Engineering,” you must add an identifier (or tag) to at least one object group name to make it unique. For example, you can use the names “Engineering-admins” and “Engineering-hosts” to make the object group names unique and to make it easier for identification.
- Additional objects can be added to an existing object group. After adding an object group, you can add more objects as required for the same group name. You do not need to re-enter existing objects; the previous configuration remains in place until the object group is removed.

- Different objects can be grouped together. For example, objects such as hosts, protocols, or services can be grouped together and configured under the same group name. Network objects can be defined only under a network group, and service objects can be defined only under a service group.
- When you define a group with the **object-group** command and use any security appliance command, the command applies to every item in that group. This feature can significantly reduce your configuration size.
- If an ACL that is associated with a class-map for ZBF inspections includes object-groups, when you add entries to or remove entries from the ACL, the changes take effect only after you exit the access-list configuration prompt.

How to Configure Object Groups for ACLs

To configure object groups for ACLs, you first create one or more object groups. These can be any combination of network object groups (groups that contain objects such as, host addresses and network addresses) or service object groups (which use operators such as **lt**, **eq**, **gt**, **neq**, and **range** with port numbers). Then, you create access control entries (ACEs) that apply a policy (such as **permit** or **deny**) to those object groups.

Creating a Network Object Group

A network object group that contains a single object (such as a single IP address, a hostname, another network object group, or a subnet) or multiple objects with a network object-group-based ACL to create access control policies for the objects.

Perform this task to create a network object group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **object-group network** *object-group-name*
4. **description** *description-text*
5. **host** {*host-address* | *host-name*}
6. *network-address* {*/nn* | *network-mask*}
7. **group-object** *nested-object-group-name*
8. Repeat the steps until you have specified objects on which you want to base your object group.
9. **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	**object-group network** *object-group-name* **Example:** `Device(config)# object-group network my-network-object-group`	Defines the object group name and enters network object-group configuration mode.
Step 4	**description** *description-text* **Example:** `Device(config-network-group)# description test engineers`	(Optional) Specifies a description of the object group. • You can use up to 200 characters.
Step 5	**host** {*host-address* | *host-name*} **Example:** `Device(config-network-group)# host 209.165.200.237`	(Optional) Specifies the IP address or name of a host. • If you specify a host address, you must use an IPv4 address.
Step 6	*network-address* {*/nn* | *network-mask*} **Example:** `Device(config-network-group)# 209.165.200.225 255.255.255.224`	(Optional) Specifies a subnet object. • You must specify an IPv4 address for the network address. The default network mask is 255.255.255.255.
Step 7	**group-object** *nested-object-group-name* **Example:** `Device(config-network-group)# group-object my-nested-object-group`	(Optional) Specifies a nested (child) object group to be included in the current (parent) object group. • The type of child object group must match that of the parent (for example, if you are creating a network object group, you must specify another network object group as the child). • You can use duplicated objects in an object group only via nesting of group objects. For example, if object 1 is in both group A and group B, you can define a group C that includes both A and B. However, you cannot include a group object that causes the group hierarchy to become circular (for example, you cannot include group A in group B and then also include group B in group A). • You can use an unlimited number of levels of nested object groups (however, a maximum of two levels is recommended).

	Command or Action	Purpose
Step 8	Repeat the steps until you have specified objects on which you want to base your object group.	—
Step 9	**end** **Example:** `Device(config-network-group)# end`	Exits network object-group configuration mode and returns to privileged EXEC mode.

Creating a Service Object Group

Use a service object group to specify TCP and/or UDP ports or port ranges. When the service object group is associated with an access control list (ACL), this service object-group-based ACL can control access to ports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **object-group service** *object-group-name*
4. **description** *description-text*
5. *protocol*
6. {**tcp** | **udp** | **tcp-udp**} [**source** {{[**eq**] | **lt** | **gt**} *port1* | **range** *port1 port2*}] [{[**eq**] | **lt** | **gt**} *port1* | **range** *port1 port2*]
7. **icmp** *icmp-type*
8. **group-object** *nested-object-group-name*
9. Repeat the steps to specify the objects on which you want to base your object group.
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	**object-group service** *object-group-name* **Example:** `Device(config)# object-group service my-service-object-group`	Defines an object group name and enters service object-group configuration mode.

	Command or Action	Purpose
Step 4	**description** *description-text* **Example:** `Device(config-service-group)# description test engineers`	(Optional) Specifies a description of the object group. • You can use up to 200 characters.
Step 5	*protocol* **Example:** `Device(config-service-group)# ahp`	(Optional) Specifies an IP protocol number or name.
Step 6	{**tcp** | **udp** | **tcp-udp**} [**source** {{[**eq**] | **lt** | **gt**} *port1* | **range** *port1 port2*}] [{[**eq**] | **lt** | **gt**} *port1* | **range** *port1 port2*] **Example:** `Device(config-service-group)# tcp-udp range 2000 2005`	(Optional) Specifies TCP, UDP, or both.
Step 7	**icmp** *icmp-type* **Example:** `Device(config-service-group)# icmp conversion-error`	(Optional) Specifies the decimal number or name of an Internet Control Message Protocol (ICMP) type.
Step 8	**group-object** *nested-object-group-name* **Example:** `Device(config-service-group)# group-object my-nested-object-group`	(Optional) Specifies a nested (child) object group to be included in the current (parent) object group. • The type of child object group must match that of the parent (for example, if you are creating a network object group, you must specify another network object group as the child). • You can use duplicated objects in an object group only via nesting of group objects. For example, if object 1 is in both group A and group B, you can define a group C that includes both A and B. However, you cannot include a group object that causes the group hierarchy to become circular (for example, you cannot include group A in group B and then also include group B in group A). • You can use an unlimited number of levels of nested object groups (however, a maximum of two levels is recommended).
Step 9	Repeat the steps to specify the objects on which you want to base your object group.	—
Step 10	**end** **Example:**	Exits service object-group configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-service-group)# end`	

Creating an Object-Group-Based ACL

When creating an object-group-based access control list (ACL), configure an ACL that references one or more object groups. As with conventional ACLs, you can associate the same access policy with one or more interfaces.

You can define multiple access control entries (ACEs) that reference object groups within the same object-group-based ACL. You can also reuse a specific object group in multiple ACEs.

Perform this task to create an object-group-based ACL.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. **remark** *remark*
5. **deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
6. **remark** *remark*
7. **permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
8. Repeat the steps to specify the fields and values on which you want to base your access list.
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	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended nomarketing`	Defines an extended IP access list using a name and enters extended access-list configuration mode.

	Command or Action	Purpose
Step 4	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark protect server by denying access from the Marketing network`	(Optional) Adds a comment about the configured access list entry. • A remark can precede or follow an access list entry. • In this example, the remark reminds the network administrator that the subsequent entry denies the Marketing network access to the interface.

Step 5	**deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
	Example:
	`Device(config-ext-nacl)# deny ip 209.165.200.244 255.255.255.224 host 209.165.200.245 log`
	`Example based on object-group:`
	`Router(config)#object-group network my_network_object_group`
	`Router(config-network-group)#209.165.200.224 255.255.255.224`
	`Router(config-network-group)#exit`
	`Router(config)#object-group network my_other_network_object_group`
	`Router(config-network-group)#host 209.165.200.245`
	`Router(config-network-group)#exit`
	`Router(config)#ip access-list extended nomarketing`
	`Router(config-ext-nacl)#deny ip object-group my_network_object_group object-group my_other_network_object_group log`
		(Optional) Denies any packet that matches all conditions specified in the statement.
		• Optionally use the **object-group** *service-object-group-name* keyword and argument as a substitute for the *protocol.* argument
		• Optionally use the **object-group** *source-network-object-group-name* keyword and argument as a substitute for the *source source-wildcard.* arguments
		• Optionally use the **object-group** *destination-network-object-group-name* keyword and argument as a substitute for the *destination destination-wildcard.* arguments
		• If the *source-wildcard* or *destination-wildcard*is omitted, a wildcard mask of 0.0.0.0 is assumed, which matches all bits of the source or destination address, respectively.
		• Optionally use the **any** keyword as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255.
		• Optionally use the **host** *source* keyword and argument to indicate a source and source wildcard of *source* 0.0.0.0 or the **host** *destination* keyword and argument to indicate a destination and destination wildcard of *destination* 0.0.0.0.
		• In this example, packets from all sources are denied access to the destination network 209.165.200.244. Logging messages about packets permitted or denied by the access list are sent to the facility configured by the **logging facility** command (for example, console, terminal, or syslog). That is, any packet that matches the access list will cause an informational logging message about the packet to be sent to the configured facility. The level of messages logged to the console is controlled by the **logging console** command.

	Command or Action	Purpose
		•
Step 6	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark allow TCP from any source to any destination`	(Optional) Adds a comment about the configured access list entry. • A remark can precede or follow an access list entry.
Step 7	**permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit tcp any any`	Permits any packet that matches all conditions specified in the statement. • Every access list needs at least one permit statement. • Optionally use the **object-group** *service-object-group-name* keyword and argument as a substitute for the *protocol.* • Optionally use the **object-group** *source-network-object-group-name* keyword and argument as a substitute for the *source source-wildcard.* • Optionally use the **object-group** *destination-network-object-group-name* keyword and argument as a substitute for the *destination destination-wildcard.* • If *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, which matches on all bits of the source or destination address, respectively. • Optionally use the **any**keyword as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • In this example, TCP packets are allowed from any source to any destination. • Use the **log-input** keyword to include input interface, source MAC address, or virtual circuit in the logging output.
Step 8	Repeat the steps to specify the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 9	**end** **Example:** `Device(config-ext-nacl)# end`	Exits extended access-list configuration mode and returns to privileged EXEC mode.

Configuring Class Maps and Policy Maps for Object Groups

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-all** *class-map-name*
4. **match access-group name** *access-list-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **pass**
9. **exit**
10. **class class-default**
11. **drop**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ogacl-cmap`	Creates a Layer 3 and Layer 4 inspect type class map and enters the class-map configuration mode.
Step 4	**match access-group name** *access-list-name* **Example:** `Device(config-cmap)# match access-group name my-ogacl-policy`	Configures a match criterion for a class map on the basis of the specified ACL.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ogacl-pmap`	Creates a inspect-type policy map and enters policy-map configuration mode.

	Command or Action	Purpose
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ogacl-cmap`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 8	**pass** **Example:** `Device(config-pmap-c)# pass`	Allows packets to be sent to a device without being inspected.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies the default class to configure or modify a policy and enters policy-map class configuration mode.
Step 11	**drop** **Example:** `Device(config-pmap-c)# drop`	Drops packets that are sent to a device.
Step 12	**end** **Example:** `Device(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Configuring Zones for Object Groups

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **interface** *type number*
8. **zone-member security** *zone-name*
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *zone-name* **Example:** `Device(config)# zone security outside`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair: a source zone and a destination zone
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 inside`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair: a source zone and a destination zone
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 8	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security inside`	Attaches an interface to a security zone.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Applying Policy Maps to Zone Pairs for Object Groups

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone-pair security** *zone-pair-name* **source** {*zone-name* | **default** | **self**} **destination** {*zone-name* | **default** | **self**}
4. **service-policy type inspect** *policy-map-name*

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	**zone-pair security** *zone-pair-name* **source** {*zone-name* | **default** | **self**} **destination** {*zone-name* | **default** | **self**} **Example:** `Device(config)# zone-pair security out-to-in source outside destination inside`	Creates a zone pair and enters security zone-pair configuration mode.
Step 4	**service-policy type inspect** *policy-map-name* **Example:** `Device(conf-sec-zone-pair)# service-policy type inspect ogacl-pmap`	Attaches a firewall policy map to a security zone pair.
Step 5	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to global configuration mode.

Verifying Object Groups for ACLs

SUMMARY STEPS

1. **enable**
2. **show object-group** [*object-group-name*]
3. **show ip access-list** [*access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show object-group** [*object-group-name*] **Example:**	Displays the configuration in the named or numbered object group (or in all object groups if no name is entered).

	Command or Action	Purpose
	`Device# show object-group my-object-group`	
Step 3	**show ip access-list** [*access-list-name*] **Example:** `Device# show ip access-list my-ogacl-policy`	Displays the contents of the named or numbered access list or object group-based ACL (or for all access lists and object group-based ACLs if no name is entered).

Configuration Examples for Object Groups for ACLs

Example: Creating a Network Object Group

The following example shows how to create a network object group named my-network-object-group, which contains two hosts and a subnet as objects:

```
Device> enable
Device# configure terminal
Device(config)# object-group network my-network-object-group
Device(config-network-group)# description test engineers
Device(config-network-group)# host 209.165.200.237
Device(config-network-group)# host 209.165.200.238

Device(config-network-group)# 209.165.200.241 255.255.255.224
Device(config-network-group)# end
```

The following example shows how to create a network object group named my-company-network, which contains two hosts, a subnet, and an existing object group (child) named my-nested-object-group as objects:

```
Device> enable
Device# configure terminal
Device(config)# object-group network my-company-network
Device(config-network-group)# host host1
Device(config-network-group)# host 209.165.200.242
Device(config-network-group)# 209.165.200.225 255.255.255.224
Device(config-network-group)# group-object my-nested-object-group
Device(config-network-group)# end
```

Example: Creating a Service Object Group

The following example shows how to create a service object group named my-service-object-group, which contains several ICMP, TCP, UDP, and TCP-UDP protocols and an existing object group named my-nested-object-group as objects:

```
Device> enable
Device# configure terminal
Device(config)# object-group service my-service-object-group
Device(config-service-group)# icmp echo
Device(config-service-group)# tcp smtp
```

```
Device(config-service-group)# tcp telnet
Device(config-service-group)# tcp source range 1 65535 telnet
Device(config-service-group)# tcp source 2000 ftp
Device(config-service-group)# udp domain
Device(config-service-group)# tcp-udp range 2000 2005
Device(config-service-group)# group-object my-nested-object-group
Device(config-service-group)# end
```

Example: Creating an Object Group-Based ACL

The following example shows how to create an object-group-based ACL that permits packets from the users in my-network-object-group if the protocol ports match the ports specified in my-service-object-group:

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended my-ogacl-policy
Device(config-ext-nacl)# permit object-group my-service-object-group object-group
my-network-object-group any
Device(config-ext-nacl)# deny tcp any any
Device(config-ext-nacl)# end
```

Example: Configuring Class Maps and Policy Maps for Object Groups

```
Device# configure terminal
Device(config)# class-map type inspect match-all ogacl-cmap
Device(config-cmap)# match access-group name my-ogacl-policy
Device(config-cmap)# exit
Device(config)# policy-map type inspect ogacl-pmap
Device(config-pmap)# class type inspect ogacl-cmap
Device(config-pmap-c)# pass
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
Device(config-pmap-c)# end
```

Example: Configuring Zones for Object Groups

```
Device# configure terminal
Device(config)# zone security outside
Device(config-sec-zone)# exit
Device(config)# zone security inside
Device(config-sec-zone)# exit
Device(config)# zone-pair security out-to-in source outside destination inside
Device(conf-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# zone-member security inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# zone-member security outside
Device(config-if)# end
```

Example: Applying Policy Maps to Zone Pairs for Object Groups

```
Device# configure terminal
Device(config)# zone-pair security out-to-in source outside destination inside
Device(config-sec-zone-pair)# service-policy type inspect ogacl-pmap
Device(config-sec-zone-pair)# end
```

Example: Verifying Object Groups for ACLs

The following example shows how to display all object groups:

```
Device# show object-group

Network object group auth-proxy-acl-deny-dest
 host 209.165.200.235
Service object group auth-proxy-acl-deny-services
 tcp eq www
 tcp eq 443
Network object group auth-proxy-acl-permit-dest
 209.165.200.226 255.255.255.224
 209.165.200.227 255.255.255.224
 209.165.200.228 255.255.255.224
 209.165.200.229 255.255.255.224
 209.165.200.246 255.255.255.224
 209.165.200.230 255.255.255.224
 209.165.200.231 255.255.255.224
 209.165.200.232 255.255.255.224
 209.165.200.233 255.255.255.224
 209.165.200.234 255.255.255.224
Service object group auth-proxy-acl-permit-services
 tcp eq www
 tcp eq 443
```

The following example shows how to display information about specific object-group-based ACLs:

```
Device# show ip access-list my-ogacl-policy

Extended IP access list my-ogacl-policy
10 permit object-group eng_service any any
```

Additional References for Object Groups for ACLs

Related Documents

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

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
ACL configuration guide	*Security Configuration Guide: Access Control Lists*

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

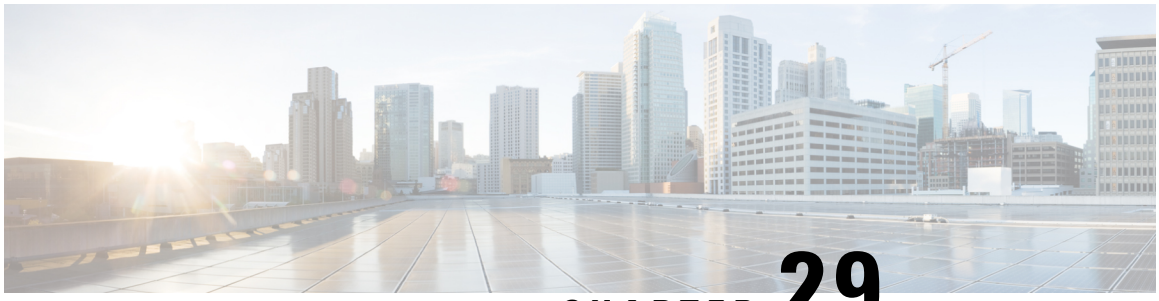
Feature Information for Object Groups for ACLs

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 37: Feature Information for Object Groups for ACLs

Feature Name	Releases	Feature Information
Object Groups for ACLs	Cisco IOS XE Release 3.12S	The Object Groups for ACLs feature lets you classify users, devices, or protocols into groups and apply them to access control lists (ACLs) to create access control policies for those groups. This feature allows multiple access control entries (ACEs), but now you can use each ACE to allow an entire group of users to access a group of servers or services or to deny them from doing so. You can use object-group ACLs with zone-based firewalls. The following commands were introduced or modified: **deny**, **ip access-group**, **ip access-list**, **object-group network**, **object-group service**, **permit**, **show ip access-list**, and **show object-group**.



CHAPTER 29

Cisco Firewall-SIP Enhancements ALG

The enhanced Session Initiation Protocol (SIP) inspection in the Cisco XE firewall provides basic SIP inspect functionality (SIP packet inspection and pinholes opening) as well as protocol conformance and application security. These enhancements give you control 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 XE software provides increased support for Cisco Call Manager, Cisco Call Manager Express, and Cisco IP-IP Gateway based voice/video systems. The application-layer gateway (ALG) SIP enhancement also supports 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 Firewall-SIP Enhancements ALG

Your system must be running Cisco IOS XE Release 2.4 or a later release.

Restrictions for Cisco Firewall-SIP Enhancements ALG

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.

Cisco ASR 1000 Series Routers

This feature was implemented without support for application inspection and control (AIC) on the Cisco ASR 1000 series routers. The Cisco IOS XE Release 2.4 supports the following commands only: **class-map type inspect**, **class type inspect**, **match protocol**, and **policy-map type inspect**.

Cisco ISR 4000 Series Routers

The Cisco IOS XE Fuji 16.7.1 release does not support Transport Layer Security (TLS) or Secure Real-time Transport Protocol (SRTP).

Information About Cisco Firewall-SIP Enhancements ALG

SIP Overview

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 that are 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 users' current locations, 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 firewall is aware of all surrounding proxies and gateways and allows the following functionalities:

- 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 UDP and the TCP format for signaling.

SIP Inspection

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

Cisco IOS XE Firewall Between SIP Phones and CCM

The Cisco IOS XE firewall is located between Cisco Call Manager or Cisco Call Manager Express and SIP phones. SIP phones are registered to Cisco Call Manager or Cisco Call Manager Express through the firewall, and any SIP calls from or to the SIP phones pass through the firewall.

Cisco IOS XE Firewall Between SIP Gateways

The Cisco IOS XE firewall is located between two SIP gateways, which can be Cisco Call Manager, Cisco Call Manager Express, 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 XE Firewall with Local Cisco Call Manager Express and Remote Cisco Call Manager Express/Cisco Call Manager

The Cisco IOS XE firewall is located between two SIP gateways, which can be Cisco Call Manager, Cisco Call Manager Express, 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 XE Firewall with Local Cisco Call Manager Express

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

ALG--SIP Over TCP Enhancement

When SIP is transferred over UDP, every SIP message is carried in one single UDP datagram. However, when SIP is transferred over TCP, one TCP segment may contain multiple SIP messages. And it is possible that the last SIP message in one of the TCP segments may be a partial one. Prior to Cisco IOS XE Release 3.5S, when there are multiple SIP messages in one received TCP segment, the SIP ALG parses only the first message. The data that is not parsed is regarded as one incomplete SIP message and returned to vTCP. When the next TCP segment is received, vTCP prefixes the unprocessed data to that segment to pass them to the SIP ALG and causes more and more data have to be buffered in vTCP.

In Cisco IOS XE Release 3.5S, the ALG--SIP over TCP Enhancement feature lets the SIP ALG to handle multiple SIP messages in one TCP segment. When a TCP segment is received, all complete SIP messages

inside this segment are parsed one-by-one. If there is an incomplete message in the end, only that portion is returned to vTCP.

How to Configure Cisco Firewall-SIP Enhancements ALG

Enabling SIP Inspection

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. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
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	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any sip-class1`	Creates an inspect type class map and enters class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol sip`	Configures the match criterion for a class map based on the named protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode.

	Command or Action	Purpose
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect sip-policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect sip-class1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies that these policy map settings apply to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 11	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can be used to troubleshoot your SIP-enabled firewall configuration:

- **clear zone-pair**
- **debug cce**
- **debug policy-map type inspect**
- **show policy-map type inspect zone-pair**
- **show zone-pair security**

Configuring a Zone Pair and Attaching a SIP Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
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* | **default**} **Example:** `Device(config)# zone security zone1`	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* | **default**} **Example:** `Device(config)# zone security zone2`	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-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]] **Example:**	Creates a zone pair and returns to security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.

	Command or Action	Purpose
	`Device(config)# zone-pair security in-out source zone1 destination zone2`	
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect sip-policy`	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 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**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 apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Cisco Firewall-SIP Enhancements ALG

Example: Enabling SIP Inspection

```
class-map type inspect match-any sip-class1
 match protocol sip
!
policy-map type inspect sip-policy
 class type inspect sip-class1
  inspect
!
class class-default
```

Example: Configuring a Zone Pair and Attaching a SIP Policy Map

```
zone security zone1
!
zone security zone2
!
zone-pair security in-out source zone1 destination zone2
 service-policy type inspect sip-policy
!
interface gigabitethernet 0/0/0
 zone security zone1
!
interface gigabitethernet 0/1/1
 zone security zone2
```

Additional References for Cisco Firewall-SIP Enhancements ALG

Related Documents

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

Related Topic	Document Title
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
Additional SIP Information	Guide to Cisco Systems VoIP Infrastructure Solution for SIP
vTCP support	*vTCP for ALG Support*

Standards and RFCs

Standard/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

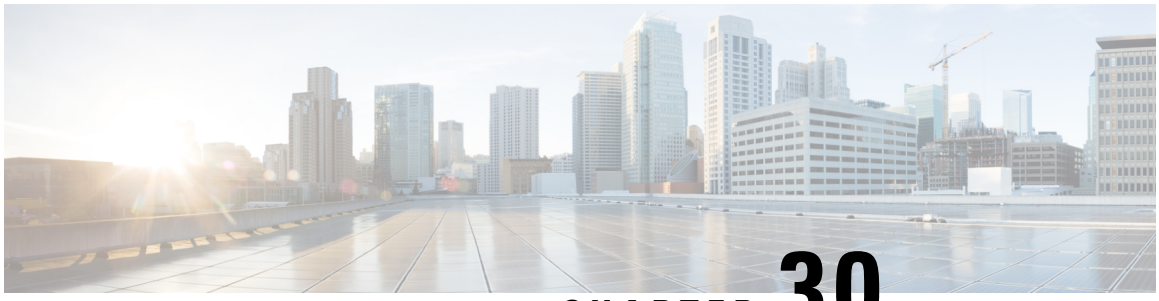
Feature Information for Cisco Firewall-SIP Enhancements ALG

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 38: Feature Information for Cisco Firewall-SIP Enhancements: ALG

Feature Name	Releases	Feature Information
AGL--SIP Over TCP Enhancement	Cisco IOS XE Release 3.5S	The ALG--SIP over TCP Enhancement feature lets the SIP ALG to handle multiple SIP messages in one TCP segment. When a TCP segment is received, all complete SIP messages inside this segment are parsed one-by-one. If there is an incomplete message in the end, only that portion is returned to vTCP.
Cisco Firewall--SIP ALG Enhancements	Cisco IOS XE Release 2.4	The Cisco Firewall--SIP ALG Enhancements feature provides voice security enhancements within the firewall feature set in Cisco IOS XE software on the Cisco ASR 1000 series routers. The following commands were implemented without support for Layer 7 (application-specific) syntax, on the Cisco ASR 1000 series routers:**class type inspect, class-map type inspect**, **match protocol**, **policy-map type inspect**.
Firewall--SIP ALG Enhancement for T.38 Fax Relay	Cisco IOS XE Release 2.4.1	The Firewall--SIP ALG Enhancement for T.38 Fax Relay feature provides an enhancement within the Firewall feature set in Cisco IOS XE software on the Cisco ASR 1000 series routers. The feature enables SIP ALG to support T.38 Fax Relay over IP, passing through the firewall on the Cisco ASR 1000 series routers.



CHAPTER 30

MSRPC ALG Support for Firewall and NAT

The MSRPC ALG Support for Firewall and NAT feature provides support for the Microsoft (MS) Remote Procedure Call (RPC) application-level gateway (ALG) on the firewall and Network Address Translation (NAT). The MSRPC ALG provides deep packet inspection (DPI) of the MSRPC protocol. The MSRPC ALG works in conjunction with a provisioning system to allow the network administrator to configure match filters to define match criteria that can be searched in an MSRPC packet.

The MSRPC ALG additionally supports the Virtual Transport Control Protocol (vTCP) functionality which provides a framework for various ALG protocols to appropriately handle the TCP segmentation and parse the segments in the Cisco IOS zone-based firewall, Network Address Translation (NAT) and other applications.

Prerequisites for MSRPC ALG Support for Firewall and NAT

- You must enable the Cisco IOS XE firewall and Network Address Translation (NAT) before applying the Microsoft (MS) Remote Procedure Call (RPC) application-level gateway (ALG) on packets.

Note MSRPC ALG is automatically enabled if traffic is sent to TCP port 135 by either Cisco IOS XE firewall or NAT, or both.

Restrictions for MSRPC ALG Support for Firewall and NAT

- Only TCP-based MSRPC is supported.
- You cannot configure the **allow** and **reset** commands together.
- You must configure the **match protocol msrpc** command for DPI.

- Only traffic that reaches destination port 135 is supported. This setting can be changed by configuration.

Information About MSRPC ALG Support for Firewall and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

MSRPC

MSRPC is a framework that developers use to publish a set of applications and services for servers and enterprises. RPC is an interprocess communication technique that allows the client and server software to communicate over the network. MSRPC is an application-layer protocol that is used by a wide array of Microsoft applications. MSRPC supports both connection-oriented (CO) and connectionless (CL) Distributed Computing Environment (DCE) RPC modes over a wide variety of transport protocols. All services of MSRPC establish an initial session that is referred to as the primary connection. A secondary session over a port range between 1024 to 65535 as the destination port is established by some services of MSRPC.

For MSRPC to work when firewall and NAT are enabled, in addition to inspecting MSRPC packets, the ALG is required to handle MSRPC specific issues like establishing dynamic firewall sessions and fixing the packet content after the NAT.

By applying MSRPC protocol inspection, most MSRPC services are supported, eliminating the need for Layer 7 policy filters.

MSRPC ALG on Firewall

After you configure the firewall to inspect the MSRPC protocol, the MSRPC ALG starts parsing MSRPC messages. The following table describes the types of Protocol Data Units (PDU) supported by the MSRPC ALG Support on Firewall and NAT feature:

Table 39: Supported PDU Types

PDU	Number	Type	Description
REQUEST	0	call	Initiates a call request.
RESPONSE	2	call	Responds to a call request.
FAULT	3	call	Indicates an RPC runtime, RPC stub, or RPC-specific exception.
BIND	11	association	Initiates the presentation negotiation for the body data.
BIND_ACK	12	association	Accepts a bind request.
BIND_NAK	13	association	Rejects an association request.
ALTER_CONTEXT	14	association	Requests additional presentation negotiation for another interface and/or version, or to negotiate a new security context, or both.
ALTER_CONTEXT_RESP	15	association	Responds to the ALTER_CONTEXT PDU. Valid values are accept or deny.
SHUTDOWN	17	call	Requests a client to terminate the connection and free the related resources.
CO_CANCEL	18	call	Cancels or orphans a connection. This message is sent when a client encounters a cancel fault.
ORPHANED	19	call	Aborts a request that in progress and that has not been entirely transmitted yet, or aborts a (possibly lengthy) response that is in progress.

MSRPC ALG on NAT

When NAT receives an MSRPC packet, it invokes the MSRPC ALG that parses the packet payload and forms a token to translate any embedded IP addresses. This token is passed to NAT, which translates addresses or ports as per your NAT configuration. The translated addresses are then written back into the packet payload by the MSRPC ALG.

If you have configured both the firewall and NAT, NAT calls the ALG first.

MSRPC Stateful Parser

The MSRPC state machine or the parser is the brain of the MSRPC ALG. The MSRPC stateful parser keeps all stateful information within the firewall or NAT depending on which feature invokes the parser first. The parser provides DPI of MSRPC protocol packets. It checks for protocol conformance and detects

out-of-sequence commands and malformed packets. As the packet is parsed, the state machine records various data and fills in the correct token information for NAT and firewall inspection.

How to Configure MSRPC ALG Support for Firewall and NAT

Note By default, MSRPC ALG is automatically enabled when NAT is enabled. There is no need to explicitly enable MSRPC ALG in the NAT-only configuration. You can use the **no ip nat service msrpc** command to disable MSRPC ALG on NAT.

Configuring a Layer 4 MSRPC Class Map and Policy Map

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. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
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 match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any msrpc-cmap`	Creates an inspect type class map for the traffic class and enters QoS class-map configuration mode.

	Command or Action	Purpose
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol msrpc`	Configures the match criteria for a class map on the basis of a specified protocol. • Only Cisco IOS XE stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect msrpc-pmap`	Creates a Layer 3 or Layer 4 inspect type policy map and enters QoS policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect msrpc-class-map`	Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 8	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS XE stateful packet inspection.
Step 9	**end** **Example:** `Router(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring a Zone Pair and Attaching an MSRPC Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *security-zone-name*
4. **exit**
5. **zone security** *security-zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** [*destination-zone*]]
8. **service-policy type inspect** *policy-map-name*

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:** `Rotuer# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *security-zone-name* **Example:** `Router(config)# zone security in-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** *security-zone-name* **Example:** `Router(config)# zone security out-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** [*destination-zone*]] **Example:** `Router(config)# zone-pair security in-out source in-zone destination out-zone`	Creates a zone pair and enters security zone pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect msrpc-pmap`	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.

	Command or Action	Purpose
Step 9	**end** **Example:** `Router(config-sec-zone-pair)# end`	Exits security zone pair configuration mode and enters privileged EXEC mode.

Enabling vTCP Support for MSRPC ALG

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **alg vtcp service msrpc**
4. **exit**
5. **set platform hardware qfp active feature alg msrpc tolerance on**

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	**alg vtcp service msrpc** **Example:** `Rotuer(config)# alg vtcp service msrpc`	Enables vTCP functionality for MSRPC ALG. **Note** By default, MSRPC ALG supports vTCP.
Step 4	**exit** **Example:** `Rotuer(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**set platform hardware qfp active feature alg msrpc tolerance on** **Example:** `Rotuer# set platform hardware qfp active feature alg msrpc tolerance on`	Enables MSRPC unknown message tolerance. **Note** By default, the tolerance is switched off.

Disabling vTCP Support for MSRPC ALG

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no alg vtcp service msrpc**
4. **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	**no alg vtcp service msrpc** **Example:** `Rotuer(config)# no alg vtcp service msrpc`	Disables vTCP functionality for MSRPC ALG.
Step 4	**end** **Example:** `Rotuer(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for MSRPC ALG Support for Firewall and NAT

Example: Configuring a Layer 4 MSRPC Class Map and Policy Map

```
Router# configure terminal
Router(config)# class-map type inspect match-any msrpc-cmap
Router(config-cmap)# match protocol msrpc
Router(config-cmap)# exit
Router(config)# policy-map type inspect msrpc-pmap
Router(config-pmap)# class type inspect msrpc-cmap
Router(config-pmap-c)# inspect
```

```
Router(config-pmap-c)# end
```

Example: Configuring a Zone Pair and Attaching an MSRPC Policy Map

```
Router# configure terminal
Router(config)# zone security in-zone
Router(config-sec-zone)# exit
Router(config)# zone security out-zone
Router(config-sec-zone)# exit
Router(config)# zone-pair security in-out source in-zone destination out-zone
Router(config-sec-zone-pair)# service-policy type inspect msrpc-pmap
Router(config-sec-zone-pair)# end
```

Example: Enabling vTCP Support for MSRPC ALG

```
Router# configure terminal
Router(config)# alg vtcp service msrpc
Router(config)# end
```

Example: Disabling vTCP Support for MSRPC ALG

```
Router# configure terminal
Router(config)# no alg vtcp service msrpc
Router(config)# end
```

Additional References for MSRPC ALG Support for Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
NAT ALGs	"Using Application-Level Gateways with NAT" module
ALG support	*NAT and Firewall ALG Support on Cisco ASR 1000 Series Routers*

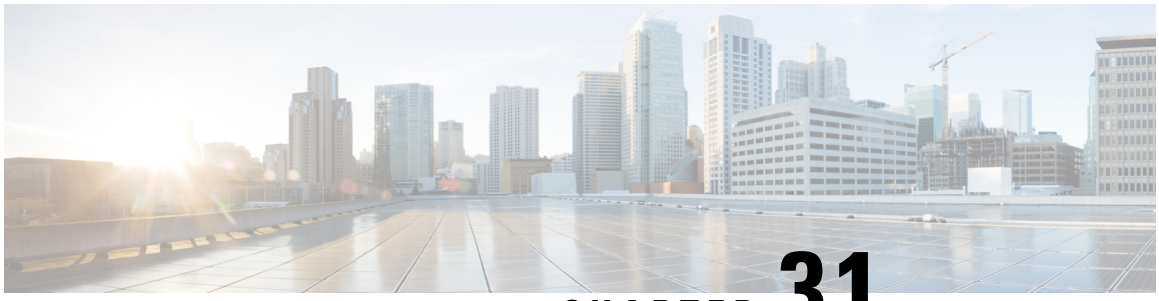
Technical Assistance

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 MSRPC ALG Support for Firewall and NAT

Table 40: Feature Information for MSRPC ALG Support for Firewall and NAT

Feature Name	Releases	Feature Information
MSRPC ALG Support for Firewall and NAT	Cisco IOS XE Release 3.5S	The MSRPC ALG Support for Firewall and NAT feature provides support for the MSRPC ALG on the firewall and NAT. The MSRPC ALG provides deep packet inspection of the MSRPC protocol. The MSRPC ALG works in conjunction with a provisioning system to allow the network administrator to configure match filters that define match criteria that can be searched in an MSRPC packet. The following commands were introduced or modified: **ip nat service msrpc**, **match protocol msrpc**.
MSRPC ALG Inspection Improvements for Zone-based Firewall and NAT	Cisco IOS XE Release 3.14S	The MSRPC ALG Inspection Improvements for Zone-based Firewall and NAT feature supports Virtual Transport Control Protocol (vTCP) functionality which provides a framework for various ALG protocols to appropriately handle the TCP segmentation and parse the segments in the Cisco firewall, Network Address Translation (NAT) and other applications. The following command was introduced: **alg vtcp service msrpc**.



CHAPTER 31

Sun RPC ALG Support for Firewalls and NAT

The Sun RPC ALG Support for Firewalls and NAT feature adds support for the Sun Microsystems remote-procedure call (RPC) application-level gateway (ALG) on the firewall and Network Address Translation (NAT). Sun RPC is an application layer protocol that enables client programs to call functions in a remote server program. This module describes how to configure the Sun RPC ALG.

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 Sun RPC ALG Support for Firewalls and NAT

- Depending on your release, the following configuration will not work on Cisco ASR 1000 Aggregation Services Routers. If you configure the inspect action for Layer 4 or Layer 7 class maps, packets that match the Port Mapper Protocol well-known port (111) pass through the firewall without the Layer 7 inspection. Without the Layer 7 inspection, firewall pinholes are not open for traffic flow, and the Sun remote-procedure call (RPC) is blocked by the firewall. As a workaround, configure the **match program-number** command for Sun RPC program numbers.
- Only Port Mapper Protocol Version 2 is supported; none of the other versions are supported.
- Only RPC Version 2 is supported.

Information About Sun RPC ALG Support for Firewalls and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

Sun RPC

The Sun remote-procedure call (RPC) application-level gateway (ALG) performs a deep packet inspection of the Sun RPC protocol. The Sun RPC ALG works with a provisioning system that allows network administrators to configure match filters. Each match filter define a match criterion that is searched in a Sun RPC packet, thereby permitting only packets that match the criterion.

In an RPC, a client program calls procedures in a server program. The RPC library packages the procedure arguments into a network message and sends the message to the server. The server, in turn, uses the RPC library and takes the procedure arguments from the network message and calls the specified server procedure. When the server procedure returns to the RPC, return values are packaged into a network message and sent back to the client.

For a detailed description of the Sun RPC protocol, see RFC 1057, *RPC: Remote Procedure Call Protocol Specification Version 2*.

Sun RPC ALG Support for Firewalls

You can configure the Sun RPC ALG by using the zone-based firewall that is created by using policies and class maps. A Layer 7 class map allows network administrators to configure match filters. The filters specify the program numbers to be searched for in Sun RPC packets. The Sun RPC Layer 7 policy map is configured as a child policy of the Layer 4 policy map with the **service-policy** command.

When you configure a Sun RPC Layer 4 class map without configuring a Layer 7 firewall policy, the traffic returned by the Sun RPC passes through the firewall, but sessions are not inspected at Layer 7. Because sessions are not inspected, the subsequent RPC call is blocked by the firewall. Configuring a Sun RPC Layer 4 class map and a Layer 7 policy allows Layer 7 inspection. You can configure an empty Layer 7 firewall policy, that is, a policy without any match filters.

Sun RPC ALG Support for NAT

By default, the Sun RPC ALG is automatically enabled when Network Address Translation (NAT) is enabled. You can use the **no ip nat service alg** command to disable the Sun RPC ALG on NAT.

How to Configure Sun RPC ALG Support for Firewalls and NAT

For Sun RPC to work when the firewall and NAT are enabled, the ALG must inspect Sun RPC packets. The ALG also handles Sun RPC-specific issues such as establishing dynamic firewall sessions and fixing the packet content after NAT translation.

Configuring the Firewall for the Sun RPC ALG

You must configure a Layer 7 Sun remote-procedure call (RPC) policy map if you have configured the inspect action for the Sun RPC protocol (that is, if you have specified the **match protocol sunrpc** command in a Layer 4 class map).

We recommend that you do not configure both security zones and inspect rules on the same interface because this configuration may not work.

Perform the following tasks to configure a firewall for the Sun RPC ALG:

Configuring a Layer 4 Class Map for a Firewall Policy

Perform this task to configure a Layer 4 class map for classifying network traffic. When you specify the **match-all** keyword with the **class-map type inspect** command, the Sun RPC traffic matches all Sun remote-procedure call (RPC) Layer 7 filters (specified as program numbers) in the class map. When you specify the **match-any** keyword with the **class-map type inspect** , the Sun RPC traffic must match at least one of the Sun RPC Layer 7 filters (specified as program numbers) in the class map.

To configure a Layer 4 class map, use the **class-map type inspect {match-any | match-all}** *classm-map-name* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
4. **match protocol** *protocol-name*
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** {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any sunrpc-l4-cmap`	Creates a Layer 4 inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol sunrpc`	Configures a match criterion for a class map on the basis of the specified protocol.
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits QoS class-map configuration mode and enters privileged EXEC mode.

Configuring a Layer 7 Class Map for a Firewall Policy

Perform this task to configure a Layer 7 class map for classifying network traffic. This configuration enables programs such as mount (100005) and Network File System (NFS) (100003) that use Sun RPC. 100005 and 100003 are Sun RPC program numbers. By default, the Sun RPC ALG blocks all programs.

For more information about Sun RPC programs and program numbers, see RFC 1057, *RPC: Remote Procedure Call Protocol Specification Version 2*.

Use the **class-map type inspect** *protocol-name* command to configure a Layer 7 class map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** *protocol-name* {**match-any** | **match-all**} *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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**class-map type inspect** *protocol-name* {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect sunrpc match-any sunrpc-l7-cmap`	Creates a Layer 7 (application-specific) inspect type class map and enters QoS class-map configuration mode.
Step 4	**match program-number** *program-number* **Example:** `Device(config-cmap)# match program-number 100005`	Specifies the allowed RPC protocol program number as a match criterion.
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits QoS class-map configuration mode and enters privileged EXEC mode.

Configuring a Sun RPC Firewall Policy Map

Perform this task to configure a Sun remote-procedure call (RPC) firewall policy map. Use a policy map to allow packet transfer for each Sun RPC Layer 7 class that is defined in a class map for a Layer 7 firewall policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *protocol-name policy-map-name*
4. **class type inspect** *protocol-name class-map-name*
5. **allow**
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** *protocol-name policy-map-name* **Example:**	Creates a Layer 7 (protocol-specific) inspect type policy map and enters QoS policy-map configuration mode.

	Command or Action	Purpose
	`Device(config)# policy-map type inspect sunrpc sunrpc-l7-pmap`	
Step 4	**class type inspect** *protocol-name class-map-name* **Example:** `Device(config-pmap)# class type inspect sunrpc sunrpc-l7-cmap`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 5	**allow** **Example:** `Device(config-pmap-c)# allow`	Allows packet transfer.
Step 6	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and returns to privileged EXEC mode.

Attaching a Layer 7 Policy Map to a Layer 4 Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class** {*class-map-name* | **class-default**}
5. **inspect** [*parameter-map-name*]
6. **service-policy** *protocol-name policy-map-name*
7. **exit**
8. **class class-default**
9. **drop**
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	**policy-map type inspect** *policy-map-name* **Example:**	Creates a Layer 4 inspect type policy map and enters QoS policy-map configuration mode.

	Command or Action	Purpose
	`Device(config)# policy-map type inspect sunrpc-l4-pmap`	
Step 4	**class** {*class-map-name* | **class-default**} **Example:** `Device(config-pmap)# class sunrpc-l4-cmap`	Associates (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 6	**service-policy** *protocol-name policy-map-name* **Example:** `Device(config-pmap-c)# service-policy sunrpc sunrpc-l7-pmap`	Attaches the Layer 7 policy map to a top-level Layer 4 policy map.
Step 7	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and returns to QoS policy-map configuration mode.
Step 8	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies the default class (commonly known as the class-default class) before you configure its policy and enters QoS policy-map class configuration mode.
Step 9	**drop** **Example:** `Device(config-pmap-c)# drop`	Configures a traffic class to discard packets belonging to a specific class.
Step 10	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class 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 the second one can be the system-defined security zone. To create the system-defined security zone or self zone, configure the **zone-pair security** command with the **self** keyword.

Note If you select a self zone, you cannot configure the inspect action.

In this task, you will do the following:

- Create security zones.
- Define zone pairs.

- Assign interfaces to security zones.
- Attach a policy map to a zone pair.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
12. **zone-member security** *zone-name*
13. **exit**
14. **interface** *type number*
15. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
16. **zone-member security** *zone-name*
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	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security z-client`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source zone and a destination zone. • In a zone pair, you can use the default zone or self zone as either the source or destination zone.
Step 4	**exit** **Example:**	Exits security zone configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-sec-zone)# exit`	
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security z-server`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source zone and a destination zone. • In a zone pair, you can use the default zone as either the source or destination zone.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name* **Example:** `Device(config)# zone-pair security clt2srv source z-client destination z-server`	Creates a zone pair and enters security zone-pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect sunrpc-l4-pmap`	Attaches a firewall policy map to a zone pair.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/0/0`	Configures an interface type and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 192.168.6.5 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 12	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z-client`	Attaches an interface to a security zone.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 14	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/1/1`	Configures an interface type and enters interface configuration mode.
Step 15	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 192.168.6.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 16	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z-server`	Attaches an interface to a security zone.
Step 17	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Sun RPC ALG Support for Firewall and NAT

Example: Configuring a Layer 4 Class Map for a Firewall Policy

```
Device# configure terminal
Device(config)# class-map type inspect match-any sunrpc-l4-cmap
Device(config-cmap)# match protocol sunrpc
Device(config-cmap)# end
```

Example: Configuring a Layer 7 Class Map for a Firewall Policy

```
Device# configure terminal
Device(config)# class-map type inspect sunrpc match-any sunrpc-l7-cmap
Device(config-cmap)# match program-number 100005
Device(config-cmap)# end
```

Example: Configuring a Sun RPC Firewall Policy Map

```
Device# configure terminal
Device(config)# policy-map type inspect sunrpc sunrpc-l7-pmap
Device(config-pmap)# class type inspect sunrpc sunrpc-l7-cmap
Device(config-pmap-c)# allow
Device(config-pmap-c)# end
```

Example: Attaching a Layer 7 Policy Map to a Layer 4 Policy Map

```
Device# configure terminal
Device(config)# policy-map type inspect sunrpcl4-pmap
Device(config-pmap)# class sunrpcl4-cmap
Device(config-pmap-c)# inspect
Device(config-pmap-c)# service-policy sunrpc sunrpc-l7-pmap
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
Device(config-pmap-c)# end
```

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

```
Device# configure terminal
Device(config)# zone security z-client
Device(config-sec-zone)# exit
Device(config)# zone security z-server
Device(config-sec-zone)# exit
Device(config)# zone-pair security clt2srv source z-client destination z-server
Device(config-sec-zone-pair)# service-policy type inspect sunrpc-l4-pmap
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 2/0/0
Device(config-if)# ip address 192.168.6.5 255.255.255.0
Device(config-if)# zone-member security z-client
Device(config-if)# exit
Device(config)# interface gigabitethernet 2/1/1
Device(config-if)# ip address 192.168.6.1 255.255.255.0
Device(config-if)# zone-member security z-server
Device(config-if)# end
```

Example: Configuring the Firewall for the Sun RPC ALG

The following is a sample firewall configuration for the Sun remote-procedure call (RPC) application-level gateway (ALG) support:

```
class-map type inspect sunrpc match-any sunrpc-l7-cmap
 match program-number 100005
!
class-map type inspect match-any sunrpc-l4-cmap
 match protocol sunrpc
!
!
policy-map type inspect sunrpc sunrpc-l7-pmap
 class type inspect sunrpc sunrpc-l7-cmap
  allow
!
!
policy-map type inspect sunrpc-l4-pmap
 class type inspect sunrpc-l4-cmap
  inspect
  service-policy sunrpc sunrpc-l7-pmap
!
 class class-default
```

```
  drop
!
!
zone security z-client
!
zone security z-server
!
zone-pair security clt2srv source z-client destination z-server
 service-policy type inspect sunrpc-l4-pmap
!
interface GigabitEthernet 2/0/0
 ip address 192.168.10.1 255.255.255.0
 zone-member security z-client
!
interface GigabitEthernet 2/1/1
 ip address 192.168.23.1 255.255.255.0
 zone-member security z-server
!
```

Additional References for Sun RPC ALG Support for Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
IP Addressing commands	IP Addressing Services Command Reference
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
RFC 1057	*RPC: Remote Procedure Call Protocol Specification Version 2*

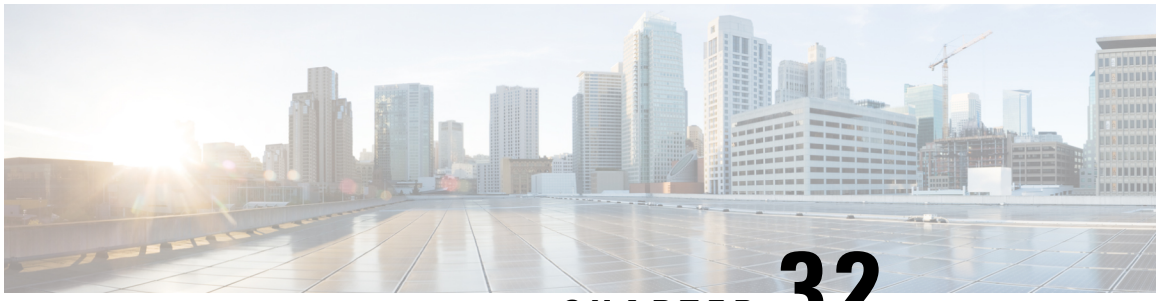
Technical Assistance

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 Sun RPC ALG Support for Firewalls and NAT

Table 41: Feature Information for Sun RPC ALG Support for Firewalls and NAT

Feature Name	Releases	Feature Information
Sun RPC ALG Support for Firewalls and NAT	Cisco IOS XE Release 3.2S	The Sun RPC ALG Support for Firewalls and NAT feature adds support for the Sun RPC ALG on the firewall and NAT. The following command was introduced or modified: **match protocol**.



CHAPTER 32

vTCP for ALG Support

Virtual Transport Control Protocol (vTCP) functionality provides a framework for various Application Layer Gateway (ALG) protocols to appropriately handle the Transport Control Protocol (TCP) segmentation and parse the segments in the Cisco firewall, Network Address Translation (NAT) and other applications.

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 vTCP for ALG Support

Your system must be running Cisco IOS XE Release 3.1 or a later Cisco IOS XE software release. The latest version of NAT or firewall ALG should be configured.

Restrictions for vTCP for ALG Support

- vTCP does not support data channel traffic. To protect system resources vTCP does not support reassembled messages larger than 8K.

- vTCP does not support the high availability functionality. High availability mainly relies on the firewall or Network Address Translation (NAT) to synchronize the session information to the standby forwarding engine.
- vTCP does not support asymmetric routing. vTCP validates and assembles packet segments based on their sequence number. If packet segments that belong to the same Layer 7 message go through different devices, vTCP will not record the proper state or do an assembly of these segments.

Information About vTCP for ALG Support

Overview of vTCP for ALG Support

When a Layer 7 protocol uses TCP for transportation, the TCP payload can be segmented due to various reasons, such as application design, maximum segment size (MSS), TCP window size, and so on. The application-level gateways (ALGs) that the firewall and NAT support do not have the capability to recognize TCP fragments for packet inspection. vTCP is a general framework that ALGs use to understand TCP segments and to parse the TCP payload.

vTCP helps applications like NAT and Session Initiation Protocol (SIP) that require the entire TCP payload to rewrite the embedded data. The firewall uses vTCP to help ALGs support data splitting between packets.

When you configure firewall and NAT ALGs, the vTCP functionality is activated.

vTCP currently supports Real Time Streaming Protocol (RTSP) and DNS ALGs.

TCP Acknowledgment and Reliable Transmission

Because vTCP resides between two TCP hosts, a buffer space is required to store TCP segments temporarily, before they are sent to other hosts. vTCP ensures that data transmission occurs properly between hosts. vTCP sends a TCP acknowledgment (ACK) to the sending host if vTCP requires more data for data transmission. vTCP also keeps track of the ACKs sent by the receiving host from the beginning of the TCP flow to closely monitor the acknowledged data.

vTCP reassembles TCP segments. The IP header and the TCP header information of the incoming segments are saved in the vTCP buffer for reliable transmission.

vTCP can make minor changes in the length of outgoing segments for NAT-enabled applications. vTCP can either squeeze the additional length of data to the last segment or create a new segment to carry the extra data. The IP header or the TCP header content of the newly created segment is derived from the original incoming segment. The total length of the IP header and the TCP header sequence numbers are adjusted accordingly.

vTCP with NAT and Firewall ALGs

ALG is a subcomponent of NAT and the firewall. Both NAT and the firewall have a framework to dynamically couple their ALGs. When the firewall performs a Layer 7 inspection or NAT performs a Layer 7 fix-up, the parser function registered by the ALGs is called and ALGs take over the packet inspection. vTCP mediates between NAT and the firewall and the ALGs that use these applications. In other words, packets are first processed by vTCP and then passed on to ALGs. vTCP reassembles the TCP segments in both directions within a TCP connection.

How to Configure vTCP for ALG Support

The RTSP, DNS, NAT, and the firewall configurations enable vTCP functionality by default. Therefore no new configuration is required to enable vTCP functionality.

Enabling RTSP on Cisco ASR 1000 Series Routers to Activate vTCP

Perform this task to enable RTSP packet inspection.

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. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **class class-default**
10. **exit**
11. **exit**
12. **zone security** *zone-name1*
13. **exit**
14. **zone security** *zone-name2*
15. **exit**
16. **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*
17. **service-policy type inspect** *policy-map-name*
18. **exit**
19. **interface** *type number*
20. **zone-member security** *zone-name1*
21. **exit**
22. **interface** *type number*
23. **zone-member security** *zone-name*
24. **end**

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 match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any rtsp_class1`	Creates an inspect type class map and enters class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol rtsp`	Configures the match criteria for a class map on the basis of the named protocol. • Use DNS in place of RTSP to configure DNS as the match protocol.
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 rtsp_policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect rtsp_class1`	Specifies the class on which the action is performed and enters policy-map-class configuration mode.
Step 8	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**class class-default** **Example:** `Router(config-pmap-c)# class class-default`	Specifies that these policy map settings apply to the predefined default class. If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 10	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy-map configuration mode.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Router(config-pmap)# exit`	Returns to global configuration mode.
Step 12	**zone security** *zone-name1* **Example:** `Router(config)# zone security private`	Creates a security zone to which interfaces can be assigned and enters security-zone configuration mode.
Step 13	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 14	**zone security** *zone-name2* **Example:** `Router(config)# zone security public`	Creates a security zone to which interfaces can be assigned and enters security-zone configuration mode.
Step 15	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 16	**zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name* **Example:** `Router(config)# zone-pair security pair-two source private destination public`	Creates a pair of security zones and enters security-zone-pair configuration mode. • To apply a policy, you must configure a zone pair.
Step 17	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy rtsp_policy`	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.
Step 18	**exit** **Example:** `Router(config-sec-zone-pair)# exit`	Returns to global configuration mode.
Step 19	**interface** *type number* **Example:** `Router(config)# GigabitEthernet0/1/0`	Specifies an interface for configuration. • Enters interface configuration mode.

	Command or Action	Purpose
Step 20	**zone-member security** *zone-name1* **Example:** `Router(config-if)# zone-member security private`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 21	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 22	**interface** *type number* **Example:** `Router(config)# GigabitEthernet0/1/0`	Specifies an interface for configuration. • Enters interface configuration mode.
Step 23	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security public`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 24	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can be used to troubleshoot your RTSP-enabled configuration:

- **clear zone-pair**
- **show policy-map type inspect zone-pair**
- **show zone-pair security**

Configuration Examples for vTCP for ALG Support

Example RTSP Configuration on Cisco ASR 1000 Series Routers

The following example shows how to configure the Cisco ASR 1000 Series Routers to enable RTSP inspection:

```
class-map type inspect match-any rtsp_class1
match protocol rtsp
policy-map type inspect rtsp_policy
class type inspect rtsp_class1
inspect
class class-default
zone security private
zone security public
zone-pair security pair-two source private destination public
service-policy type inspect rtsp_policy
interface GigabitEthernet0/1/0
 ip address 10.0.0.1 255.0.0.0
zone-member security private
!
interface GigabitEthernet0/1/1
 ip address 10.0.1.1 255.0.0.0
  zone-member security public
```

Additional References for vTCP for ALG Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS 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
Cisco Firewall--SIP Enhancements: ALG	*Security Configuration Guide: Securing the Data Plane*
Network Address Translation	*IP Addressing Services Configuration*

Standards and RFCs

Standard/RFC	Title
RFC 793	*Transport Control Protocol*
RFC 813	*Window and Acknowledge Strategy in TCP*

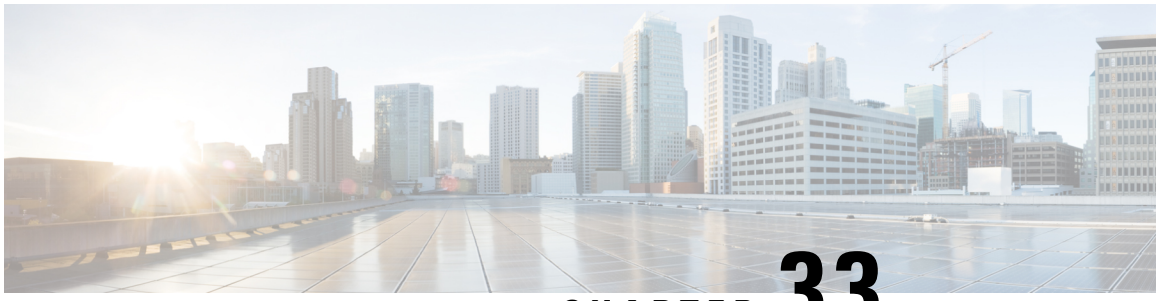
Technical Assistance

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 vTCP for ALG Support

Table 42: Feature Information for vTCP for ALG Support

Feature Name	Releases	Feature Information
vTCP for ALG Support	Cisco IOS XE Release 3.1S	This functionality provides an enhancement to handle the TCP segmentation and reassembling for the firewall and NAT ALGs, in Cisco IOS XE software on the Cisco ASR 1000 Series Routers.



CHAPTER 33

ALG—H.323 vTCP with High Availability Support for Firewall and NAT

The ALG—H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 application-level gateway (ALG) to support a TCP segment that is not a single H.323 message. Virtual TCP (vTCP) supports TCP segment reassembly. Prior to this introduction of the feature, the H.323 ALG processed a TCP segment only if it was a complete H.323 message. If the TCP segment was more than one message, the H.323 ALG ignored the TCP segment and the packet was passed without processing.

This module describes how to configure the ALG—H.323 vTCP with high availability (HA) support for 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.

Restrictions for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

- When an incoming TCP segment is not a complete H.323 message, the H.323 ALG buffers the TCP segment while waiting for the rest of the message. The buffered data is not synchronized to the standby device for high availability (HA).
- The performance of the H.323 ALG may get impacted when vTCP starts to buffer data.

Information About ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

Basic H.323 ALG Support

H.323 is a recommendation published by the ITU-T defining a series of network elements and protocols for multimedia transmission through packet-based networks. H.323 defines a number of network elements used in multimedia transmission.

Although most H.323 implementations today utilize TCP as the transport mechanism for signaling, H.323 Version 2 enables basic UDP transport.

- H.323 Terminal—This element is an endpoint in the network, providing two-way communication with another H.323 terminal or gateway.
- H.323 Gateway—This element provides protocol conversion between H.323 terminals and other terminals that do not support H.323.
- H.323 Gatekeeper—This element provides services like address translation, network access control, and bandwidth management and account for H.323 terminals and gateways.

The following core protocols are described by the H.323 specification:

- H.225—This protocol describes call signaling methods used between any two H.323 entities to establish communication.
- H.225 Registration, Admission, and Status (RAS)—This protocol is used by the H.323 endpoint and gateway for address resolution and admission control services.
- H.245—This protocol is used for exchanging the capabilities of multimedia communication and for the opening and closing of logical channels for audio, video, and data.

In addition to the protocols listed, the H.323 specification describes the use of various IETF protocols like the Real Time Transport (RTP) protocol and audio (G.711, G.729, and so on) and video (H.261, H.263, and H.264) codecs.

NAT requires a variety of ALGs to handle Layer 7 protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels. The H.323 ALG performs these specific services for H.323 messages.

Overview of vTCP for ALG Support

When a Layer 7 protocol uses TCP for transportation, the TCP payload can be segmented due to various reasons, such as application design, maximum segment size (MSS), TCP window size, and so on. The application-level gateways (ALGs) that the firewall and NAT support do not have the capability to recognize TCP fragments for packet inspection. vTCP is a general framework that ALGs use to understand TCP segments and to parse the TCP payload.

vTCP helps applications like NAT and Session Initiation Protocol (SIP) that require the entire TCP payload to rewrite the embedded data. The firewall uses vTCP to help ALGs support data splitting between packets.

When you configure firewall and NAT ALGs, the vTCP functionality is activated.

vTCP currently supports Real Time Streaming Protocol (RTSP) and DNS ALGs.

TCP Acknowledgment and Reliable Transmission

Because vTCP resides between two TCP hosts, a buffer space is required to store TCP segments temporarily, before they are sent to other hosts. vTCP ensures that data transmission occurs properly between hosts. vTCP sends a TCP acknowledgment (ACK) to the sending host if vTCP requires more data for data transmission. vTCP also keeps track of the ACKs sent by the receiving host from the beginning of the TCP flow to closely monitor the acknowledged data.

vTCP reassembles TCP segments. The IP header and the TCP header information of the incoming segments are saved in the vTCP buffer for reliable transmission.

vTCP can make minor changes in the length of outgoing segments for NAT-enabled applications. vTCP can either squeeze the additional length of data to the last segment or create a new segment to carry the extra data. The IP header or the TCP header content of the newly created segment is derived from the original incoming segment. The total length of the IP header and the TCP header sequence numbers are adjusted accordingly.

vTCP with NAT and Firewall ALGs

ALG is a subcomponent of NAT and the firewall. Both NAT and the firewall have a framework to dynamically couple their ALGs. When the firewall performs a Layer 7 inspection or NAT performs a Layer 7 fix-up, the parser function registered by the ALGs is called and ALGs take over the packet inspection. vTCP mediates

between NAT and the firewall and the ALGs that use these applications. In other words, packets are first processed by vTCP and then passed on to ALGs. vTCP reassembles the TCP segments in both directions within a TCP connection.

Overview of ALG—H.323 vTCP with High Availability Support

The ALG-H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 application-level gateway (ALG) to support a TCP segment that is not a single H.323 message. After the H.323 ALG is coupled with vTCP, the firewall and NAT interact with the H.323 ALG through vTCP. When vTCP starts to buffer data, the high availability (HA) function is impacted, because vTCP cannot synchronize the buffered data to a standby device. If the switchover to the standby device happens when vTCP is buffering data, the connection may be reset if the buffered data is not synchronized to the standby device. After the buffered data is acknowledged by vTCP, the data is lost and the connection is reset. The firewall and NAT synchronize the data for HA. vTCP only synchronizes the status of the current connection to the standby device, and in case of errors, the connection is reset.

How to Configure ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Configuring ALG—H.323 vTCP with High Availability Support for Firewalls

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **match protocol** *protocol-name*
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class type inspect** *class-map-name*
9. **inspect**
10. **exit**
11. **class class-default**
12. **exit**
13. **zone security** *zone-name*
14. **exit**
15. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
16. **service-policy type inspect** *policy-map-name*
17. **exit**
18. **interface** *type number*
19. **zone member security** *zone-name*
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.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any h.323-class`	Creates an inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol h323`	Configures the match criteria for a class map on the basis of the named protocol.
Step 5	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol h323ras`	Configures the match criteria for a class map on the basis of the named protocol.
Step 6	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect h.323-policy`	Creates an inspect type policy map and enters QoS policy-map configuration mode.
Step 8	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect h.323-class`	Specifies the class on which the action is performed and enters QoS policy-map class configuration mode.
Step 9	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 10	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters policy-map configuration mode.

	Command or Action	Purpose
Step 11	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Applies the policy map settings to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 12	**exit** **Example:** `Device(config)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 13	**zone security** *zone-name* **Example:** `Device(config)# zone security inside`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source and a destination zone. • In a zone pair, you can use the default zone as either the source or the destination zone.
Step 14	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 15	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security inside-outside source inside destination outside`	Creates a pair of security zones and enters security-zone-pair configuration mode. • To apply a policy, you must configure a zone pair.
Step 16	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect h.323-policy`	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.
Step 17	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 18	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 19	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security inside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the

	Command or Action	Purpose
		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 20	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Example: Configuring ALG—H.323 vTCP with High Availability Support for Firewalls

```
Device# configure terminal
Device(config)# class-map type inspect h.323-class
Device(config-cmap)# match protocol h323
Device(config-cmap)# match protocol h323ras
Device(config-cmap)# exit
Device(config)# policy-map type inspect h323-policy
Device(config-pmap)# class type inspect h323
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap)# exit
Device(config)# zone security inside
Device(config-sec-zone)# exit
Device(config)# zone security outside
Device(config-sec-zone)# exit
Device(config)# zone-pair security inside-outside source inside destination outside
Device(config-sec-zone-pair)# service-policy type inspect h.323-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# zone-member security inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# zone-member security outside
Device(config-if)# end
```

Additional References for ALG-H.323 vTCP with High Availability Support for Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
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
NAT commands	IP Addressing Services Command Reference

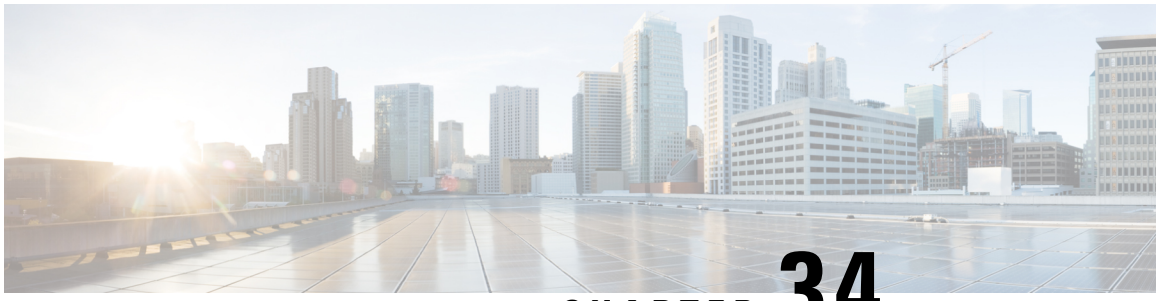
Technical Assistance

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 ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Table 43: Feature Information for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Feature Name	Releases	Feature Information
ALG—H.323 vTCP with High Availability Support for Firewall and NAT	Cisco IOS XE Release 3.7S	The ALG—H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 ALG to support a TCP segment that is not a single H.323 message. vTCP supports segment reassembly. Prior to the introduction of this feature, the H.323 ALG processed a TCP segment only if it was a complete H.323 message. If the TCP segment was more than one message, the H.323 ALG ignored the TCP segment and the packet was passed without processing.



CHAPTER 34

FTP66 ALG Support for IPv6 Firewalls

The FTP66 ALG Support for IPv6 Firewalls feature allows FTP to work with IPv6 firewalls. This module describes how to configure a firewall, Network Address Translation (NAT), and Stateful NAT64 to work with the FTP66 application-level gateway (ALG).

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 FTP66 ALG Support for IPv6 Firewalls

The FTP66 ALG does not support the following:

- Box-to-box high availability.
- Per-subscriber firewalls.
- Stateless Network Address Translation 64 (NAT64).
- Virtual routing and forwarding (VRF) when stateful NAT64 is configured.
- Virtual TCP (vTCP) or the breaking up of packets into smaller packets after translation.

Information About FTP66 ALG Support for IPv6 Firewalls

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

FTP66 ALG Support Overview

Firewalls support the inspection of IPv6 packets and stateful Network Address Translation 64 (NAT64). For FTP to work over IPv6 packet inspection, the application-layer gateway (ALG) (also called the application-level gateway [ALG]), FTP66, is required. The FTP66 ALG is also called all-in-one FTP ALG and one FTP ALG.

The FTP66 ALG supports the following:

- Firewall IPv4 packet inspection
- Firewall IPv6 packet inspection
- NAT configuration
- NAT64 configuration (along with FTP64 support)
- NAT and firewall configuration
- NAT64 and firewall configuration

The FTP66 ALG has the following security vulnerabilities:

- Packet segmentation attack—The FTP ALG state machine can detect segmented packets, and the state machine processing is stopped until a complete packet is received.
- Bounce attack—The FTP ALG does not create doors (for NAT) or pinholes (for firewalls) with a data port number less than 1024. The prevention of a bounce attack is activated only when the firewall is enabled.

FTP Commands Supported by FTP66 ALG

The FTP66 application-level gateway (ALG) is based on RFC 959. This section describes the main RFC 959 and RFC 2428 FTP commands and responses that the FTP66 ALG processes.

PORT Command

The PORT command is used in active FTP mode. The PORT command specifies the address and the port number to which a server should connect. When you use this command, the argument is a concatenation of a 32-bit Internet host address and a 16-bit TCP port address. The address information is broken into 8-bit fields, and the value of each field is transmitted as a decimal number (in character string representation). The fields are separated by commas.

The following is a sample PORT command, where *h1* is the highest order 8-bit of the Internet host address:

```
PORT h1,h2,h3,h4,p1,p2
```

PASV Command

The PASV command requests a server to listen on a data port that is not the default data port of the server and to wait for a connection, rather than initiate another connection, when a TRANSFER command is received. The response to the PASV command includes the host and port address the server is listening on.

Extended FTP Commands

Extended FTP commands provide a method by which FTP can communicate the data connection endpoint information for network protocols other than IPv4. Extended FTP commands are specified in RFC 2428. In RFC 2428, the extended FTP commands EPRT and EPSV, replace the FTP commands PORT and PASV, respectively.

EPRT Command

The EPRT command allows you to specify an extended address for data connection. The extended address must consist of a network protocol, network address, and transport address. The format of an EPRT command is as follows:

```
EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>
```

- The <net-prt> argument must be an address family number and must be defined as described in the table below.

  Table 44: The <net-prt> Argument Definitions

Address Family Number	Protocol
1	IPv4 (Pos81a)
2	IPv6 (DH96)

- The <net-addr> argument is a protocol-specific string representation of the network address. For the two address family numbers specified in the table above (address family numbers 1 and 2), the addresses must be in the format listed in the table below.

Address Family Number	Address Format	Example
1	Dotted decimal	10.135.1.2
2	IPv6 string representations defined in DH96	2001:DB8:1::1

- The <tcp-port> argument must be a string representation of the number of the TCP port on which the host is listening for data connection.
- The following command shows how to specify the server to use an IPv4 address to open a data connection to host 10.235.1.2 on TCP port 6275:

```
EPRT |1|10.235.1.2|6275|
```

- The following command shows how to specify the server to use an IPv6 network protocol and a network address to open a TCP data connection on port 5282:

```
EPRT |2|2001:DB8:2::2:417A|5282|
```

- The <d> argument is the delimiter character and it must be in ASCII format, in the range from 33 to 126.

EPSV Command

The EPSV command requests that a server listen on a data port and wait for a connection. The response to this command includes only the TCP port number of the listening connection. The response code for entering passive mode by using an extended address must be 229.

The text returned in response to an EPSV command must be in the following format:

```
(<d><d><d><tcp-port><d>)
```

- The portion of the string enclosed in parentheses must be the exact string needed by the EPRT command to open the data connection.

  The first two fields in parentheses must be blank. The third field must be a string representation of the TCP port number on which the server is listening for a data connection. The network protocol used by the data connection is the same network protocol used by the control connection. The network address used to establish the data connection is the same network address used for the control connection.

- The following is a sample response string:

```
Entering Extended Passive Mode (|||6446|)
```

The following FTP responses and commands are also processed by the FTP66 ALG. The results of processing these commands are used to drive the transition in the state machine.

- 230 response
- AUTH
- USER
- PASS

How to Configure FTP66 ALG Support for IPv6 Firewalls

Configuring a Firewall for FTP66 ALG Support

You need to explicitly enable the FTP66 ALG by using the **match protocol ftp** command.

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. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
11. **exit**
12. **exit**
13. **zone security** *zone-name*
14. **exit**
15. **zone-pair security** *zone-pair* **source** *source-zone* **destination** *destination-zone*
16. **service-policy type inspect** *policy-map-name*
17. **exit**
18. **interface** *type number*
19. **no ip address**
20. **ip virtual-reassembly**
21. **zone-member security** *zone-name*
22. **negotiation auto**
23. **ipv6 address** *ipv6-address/prefix-length*
24. **cdp enable**
25. **exit**
26. **ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number*
27. **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*
28. **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 match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any in2out-class`	Creates an inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol ftp`	Configures a match criteria for a class map on the basis of the named protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect in-to-out`	Creates an inspect type policy map and enters QoS policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect in2out-class`	Specifies the class on which an action is performed and enters QoS policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Applies the policy map settings to the predefined default class and enters QoS policy-map class configuration mode. • If the traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 11	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config-pmap)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 13	**zone security** *zone-name* **Example:** `Device(config)# zone security inside`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source and a destination zone. • In a zone pair, you can use the default zone as either the source or the destination zone.
Step 14	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 15	**zone-pair security** *zone-pair* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security in2out source inside destination outside`	Creates a pair of security zones and enters security zone-pair configuration mode. • To apply a policy, you must configure a zone pair.
Step 16	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect in-to-out`	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.
Step 17	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 18	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 19	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 20	**ip virtual-reassembly** **Example:** `Device(config-if)# ip virtual-reassembly`	Enables virtual fragmentation reassembly (VFR) on an interface.

	Command or Action	Purpose
Step 21	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security inside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into 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 apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 22	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 23	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 24	**cdp enable** **Example:** `Device(config-if)# cdp enable`	Enables Cisco Discovery Protocol on an interface.
Step 25	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 26	**ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001::/96 gigabitethernet 0/0/1`	Establishes static IPv6 routes.
Step 27	**ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address* **Example:** `Device(config)# ipv6 neighbor 2001:DB8:1::1 gigabitethernet 0/0/1 0000.29f1.4841`	Configures a static entry in the IPv6 neighbor discovery cache.
Step 28	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring NAT for FTP66 ALG Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat inside**
6. **zone-member security** *zone-name*
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat outside**
11. **zone-member security** *zone-name*
12. **exit**
13. **ip nat inside source static** *local-ip global-ip*
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 terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/2`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Indicates that an interface is connected to the inside network (the network that is subject to NAT translation).
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security inside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except

	Command or Action	Purpose
		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 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.2.1.1 255.255.255.0`	Indicates that an interface is connected to the inside network (the network that is subject to NAT translation).
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Indicates that the interface is connected to the outside network.
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security outside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into 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 apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 13	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 10.1.1.10 10.1.1.80`	Enables NAT of the inside source address.
Step 14	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring NAT64 for FTP66 ALG Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **no ip address**
6. **ipv6 virtual-reassembly**
7. **zone-member security** *zone-name*
8. **negotiation auto**
9. **ipv6 address** *ipv6-address*
10. **ipv6 enable**
11. **nat64 enable**
12. **cdp enable**
13. **exit**
14. **interface** *type number*
15. **ip address** *type number*
16. **ip virtual-reassembly**
17. **zone member security** *zone-name*
18. **negotiation auto**
19. **nat64 enable**
20. **exit**
21. **ipv6 route** *ipv6-address interface-type interface-number*
22. **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*
23. **nat64 v6v4 static** *ipv6-address ipv4-address*
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.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 5	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 6	**ipv6 virtual-reassembly** **Example:** `Device(config-if)# ipv6 virtual-reassembly`	Enables virtual fragmentation reassembly (VFR) on an interface.
Step 7	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security inside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into 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 apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 8	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 9	**ipv6 address** *ipv6-address* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::2/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 10	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 11	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 on an interface.
Step 12	**cdp enable** **Example:** `Device(config-if)# cdp enable`	Enables Cisco Discovery Protocol on an interface.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 14	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 15	**ip address** *type number* **Example:** `Device(config-if)# ip address 209.165.201.25 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 16	**ip virtual-reassembly** **Example:** `Device(config-if)# ip virtual-reassembly`	Enables VFR on an interface.
Step 17	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security outside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 18	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 19	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 on an interface.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 21	**ipv6 route** *ipv6-address interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0`	Establishes static IPv6 routes and specifies the IPv6 address of the next hop that can be used to reach a specified network.
Step 22	**ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address* **Example:** `Device(config)# ipv6 neighbor 2001:DB8:1::103 gigabitethernet 0/0/0 0000.29f1.4841`	Configures a static entry in the IPv6 neighbor discovery cache.

	Command or Action	Purpose
Step 23	**nat64 v6v4 static** *ipv6-address ipv4-address* **Example:** `Device(config)# nat64 v6v4 static 2001:DB8:1::103 209.165.201.32`	Translates an IPv6 source address to an IPv4 source address and an IPv4 destination address to an IPv6 destination address for NAT64.
Step 24	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for FTP66 ALG Support for IPv6 Firewalls

Example: Configuring an IPv6 Firewall for FTP66 ALG Support

```
Device# configure terminal
Device(config)# class-map type inspect match-any in2out-class
Device(config-cmap)# match protocol ftp
Device(config-cmap)# exit
Device(config)# policy-map type inspect in-to-out
Device(config-pmap)# class type inspect in2out-class
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# zone security inside
Device(config-sec-zone)# exit
Device(config)# zone security outside
Device(config-sec-zone)# exit
Device(config)# zone-pair security in2out source inside destination outside
Device(config-sec-zone-pair)# service-policy type inspect in-to-out
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# ip virtual-reassembly
Device(config-if)# zone-member security inside
Device(config-if)# negotiation auto
Device(config-if)# ipv6 address 2001:DB8:1::1/96
Device(config-if)# cdp enable
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# no ip address
Device(config-if)# ip virtual-reassembly
Device(config-if)# zone-member security outside
Device(config-if)# negotiation auto
Device(config-if)# ipv6 address 2001:DB8:2::2/96
Device(config-if)# exit
Device(config)# ipv6 route 2001::/96 gigabitethernet 0/0/1
Device(config)# ipv6 route 2001::/96 gigabitethernet 0/1/1
Device(config)# ipv6 neighbor 2001:DB8:1::1 gigabitethernet 0/0/1 0000.29f1.4841
Device(config)# ipv6 neighbor 2001:DB8:2::2 gigabitethernet 0/1/1 0000.29f1.4842
Device(config)# end
```

Example: Configuring NAT for FTP66 ALG Support

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1/2
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# zone-member security inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# ip address 10.2.1.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# zone-member security outside
Device(config-if)# exit
Device(config-if)# ip nat inside source static 10.1.1.10 10.1.1.80
```

Example: Configuring NAT64 for FTP66 ALG Support

```
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# no ip address
Device(config-if)# ipv6 virtual-reassembly
Device(config-if)# zone-member security inside
Device(config-if)# negotiation auto
Device(config-if)# ipv6 address 2001:DB8:1::2/96
Device(config-if)# ipv6 enable
Device(config-if)# nat64 enable
Device(config-if)# cdp enable
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# ip address 209.165.201.25 255.255.255.0
Device(config-if)# ip virtual-reassembly
Device(config-if)# zone member security outside
Device(config-if)# negotiation auto
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0
Device(config)# 2001:DB8:1::103 gigabitethernet 0/0/0 0000.29f1.4841
Device(config)# nat64 v6v4 static 2001:DB8:1::103 209.165.201.32
```

Additional References for FTP66 ALG Support for IPv6 Firewalls

Related Documents

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

Related Topic	Document Title
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
NAT commands	IP Addressing Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 959	*File Transfer Protocol*
RFC 2428	*FTP Extensions for IPv6 and NATs*

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

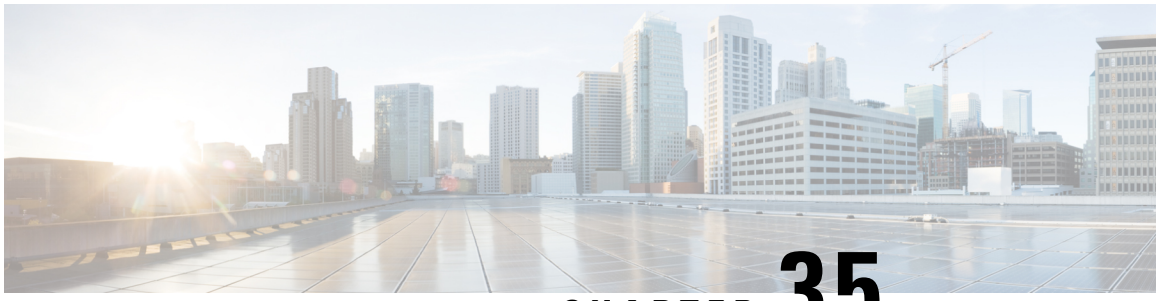
Feature Information for FTP66 ALG Support for IPv6 Firewalls

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 45: Feature Information for FTP66 ALG Support for IPv6 Firewalls

Feature Name	Releases	Feature Information
FTP66 ALG Support for IPv6 Firewalls	Cisco IOS XE Release 3.7S	The FTP66 ALG Support for IPv6 Firewalls feature allows FTP to work with IPv6 firewalls. This module describes how to configure a firewall, Network Address Translation (NAT), and NAT64 to work with the FTP66 application-level gateway (ALG).



CHAPTER 35

SIP ALG Hardening for NAT and Firewall

The SIP ALG Hardening for NAT and Firewall feature provides better memory management and RFC compliance over the existing Session Initiation Protocol (SIP) application-level gateway (ALG) support for Network Address Translation (NAT) and firewall. This feature provides the following enhancements:

- Management of the local database for all SIP Layer 7 data
- Processing of the Via header
- Support for logging additional SIP methods
- Support for Provisional Response Acknowledgment (PRACK) call flow
- Support for the Record-Route header

The above enhancements are available by default; no additional configuration is required on NAT or firewall.

This module explains the SIP ALG enhancements and describes how to enable NAT and firewall support for SIP.

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 SIP ALG Hardening for NAT and Firewall

- Session Initiation Protocol (SIP) application-level gateway (ALG) does not provide any security features.
- SIP ALG manages the local database based on call IDs. There might be a corner case involving two calls coming from two different clients with the same call ID, resulting in call ID duplication.

Information About SIP ALG Hardening for NAT and Firewall

SIP Overview

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 that are 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 users' current locations, 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.

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

SIP ALG Local Database Management

A Session Initiation Protocol (SIP) trunk is a direct connection of an IP PBX to a service provider over an IP network using SIP. There can be numerous concurrent calls in a SIP trunk. During the call setup process, all calls use the same control channel for call establishment. More than one call uses the same control channel for call setup. When the same control channel is used by more than one call, the stateful information stored

in the control-channel sessions becomes unreliable. SIP stateful information consists of media channel information such as the IP address and port number used by client and server endpoints to send media data. The media channel information is used to create a firewall pinhole and a Network Address Translation (NAT) door for the data channel in firewall and NAT, respectively. Because multiple calls use the same control channel for call setup, there will be multiple sets of media data.

In a SIP trunk, more than one call shares the same firewall and NAT session. NAT and firewall identify and manage a SIP session by using the 5 tuple in a SIP packet—source address, destination address, source port, destination port, and protocol. The conventional method of using the 5 tuple to identify and match calls does not completely support SIP trunking and often leads to Layer 7 data memory leaks and call matching issues.

In contrast to other application-level gateways (ALGs), SIP ALG manages the SIP Layer 7 data by using a local database to store all media-related information contained in normal SIP calls and in SIP calls embedded in a SIP trunk. SIP ALG uses the Call-ID header field contained in a SIP message to search the local database for call matching and to manage and terminate calls. The Call-ID header field is a dialog identifier that identifies messages belonging to the same SIP dialog.

SIP ALG uses the call ID to perform search in the local database and to manage memory resources. In certain scenarios where SIP ALG is unable to free up a Layer 7 data record from the database, a session timer is used to manage and free resources to ensure that there are no stalled call records in the database.

Note Because all Layer 7 data is managed by SIP ALG by using a local database, SIP ALG never replies on firewall and NAT to free SIP Layer 7 data; SIP ALG frees the data by itself. If you use the **clear** command to clear all NAT translations and firewall sessions, the SIP Layer 7 data in the local database is not freed.

SIP ALG Via Header Support

A Session Initiation Protocol (SIP) INVITE request contains a Via header field. The Via header field indicates the transport paths taken by a SIP request. The Via header also contains information about the return path for subsequent SIP responses, which includes the IP address and the port to which the response message is to be sent.

SIP ALG creates a firewall pinhole or a Network Address Translation (NAT) door based on the first value in the Via header field for each SIP request received, except the acknowledge (ACK) message. If the port number information is missing from the first Via header, the port number is assumed to be 5060.

SIP ALG Method Logging Support

The SIP ALG Hardening for NAT and Firewall feature provides support for detailed logging of the following methods in Session Initiation Protocol (SIP) application-level gateway (ALG) statistics:

- PUBLISH
- OPTIONS
- 1XX (excluding 100,180,183)
- 2XX (excluding 200)

The existing SIP methods that are logged in SIP ALG statistics include ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, REFER, REGISTER, SUBSCRIBE, and 1XX-6XX.

SIP ALG PRACK Call-Flow Support

Session Initiation Protocol (SIP) defines two types of responses: final and provisional. Final responses convey the result of processing a request and are sent reliably. Provisional responses, on the other hand, provide information about the progress of processing a request but are not sent reliably.

Provisional Response Acknowledgement (PRACK) is a SIP method that provides an acknowledgment (ACK) system for provisional responses. PRACK allows reliable exchanges of SIP provisional responses between SIP endpoints. SIP reliable provisional responses ensure that media information is exchanged and resource reservation can occur before connecting the call.

SIP uses the connection, media, and attribute fields of the Session Description Protocol (SDP) during connection negotiation. SIP application-level gateway (ALG) supports SDP information within a PRACK message. If media information exists in a PRACK message, SIP ALG retrieves and processes the media information. SIP ALG also handles the creation of media channels for subsequent media streams. SIP ALG creates a firewall pinhole and a NAT door based on the SDP information in PRACK messages.

SIP ALG Record-Route Header Support

The Record-Route header field is added by a Session Initiation Protocol (SIP) proxy to a SIP request to force future requests in a SIP dialog to be routed through the proxy. Messages sent within a dialog then traverse all SIP proxies, which add a Record-Route header field to the SIP request. The Record-Route header field contains a globally reachable Uniform Resource Identifier (URI) that identifies the proxy.

SIP application-level gateway (ALG) parses the Contact header and uses the IP address and the port value in the Contact header to create a firewall pinhole and a Network Address Translation (NAT) door. In addition, SIP ALG supports the parsing of the Record-Route header to create a firewall pinhole and a NAT door for future messages that are routed through proxies.

With the parsing of the Record-Route header, SIP ALG supports the following scenarios:

- A Cisco ASR 1000 Aggregation Services Router is deployed between two proxies.
- A Cisco ASR 1000 Aggregation Services Router is deployed between a User Agent Client (UAC) and a proxy.
- A Cisco ASR 1000 Aggregation Services Router is deployed between a proxy and a User Agent Server (UAS).
- No proxy exists between the client and the server. No record routing occurs in this scenario.

How to Configure SIP ALG Hardening for NAT and Firewall

Enabling NAT for SIP Support

NAT support for SIP is enabled by default on port 5060. If this feature has been disabled, perform this task to re-enable NAT support for SIP. To disable the NAT support for SIP, use the **no ip nat service sip** command.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip nat service sip** {**tcp** | **udp**} **port** *port-number*
4. **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 nat service sip** {**tcp**	**udp**} **port** *port-number* **Example:** `Device(config)# ip nat service sip tcp port 5060`	Enables NAT support for SIP.
Step 4	**end** **Example:** `Device(config)# end`	Exist global configuration mode and returns to privileged EXEC mode.

Enabling SIP Inspection

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. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
11. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device> enable`	
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 sip-class1`	Creates an inspect type class map and enters class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol sip`	Configures the match criterion for a class map based on the named protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect sip-policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect sip-class1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies that these policy map settings apply to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 11	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring a Zone Pair and Attaching a SIP Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
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* | **default**} **Example:** `Device(config)# zone security zone1`	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* | **default**} **Example:** `Device(config)# zone security zone2`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]] **Example:** `Device(config)# zone-pair security in-out source zone1 destination zone2`	Creates a zone pair and returns to security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect sip-policy`	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 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**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 apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone2`	Assigns an interface to a specified security zone.

	Command or Action	Purpose
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for SIP ALG Hardening for NAT and Firewall

Example: Enabling NAT for SIP Support

```
Device> enable
Device# configure terminal
Device(config)# ip nat service sip tcp port 5060
Device(config)# end
```

Example: Enabling SIP Inspection

```
class-map type inspect match-any sip-class1
 match protocol sip
!
policy-map type inspect sip-policy
 class type inspect sip-class1
  inspect
!
class class-default
```

Example: Configuring a Zone Pair and Attaching a SIP Policy Map

```
zone security zone1
!
zone security zone2
!
zone-pair security in-out source zone1 destination zone2
 service-policy type inspect sip-policy
!
interface gigabitethernet 0/0/0
 zone security zone1
!
interface gigabitethernet 0/1/1
 zone security zone2
```

Additional References for SIP ALG Hardening for NAT and Firewall

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT configuration	*IP Addressing: NAT Configuration Guide*
Firewall configuration	*Security Configuration Guide: Zone-Based Policy Firewall*
NAT commands	Cisco IOS IP Addressing Services Command Reference
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
NAT and firewall ALG support	NAT and Firewall ALG and AIC Support on Cisco ASR 1000 Series Aggregation Services Routers matrix

Standards and RFCs

Standard/RFC	Title
RFC 3261	*SIP: Session Initiation Protocol*

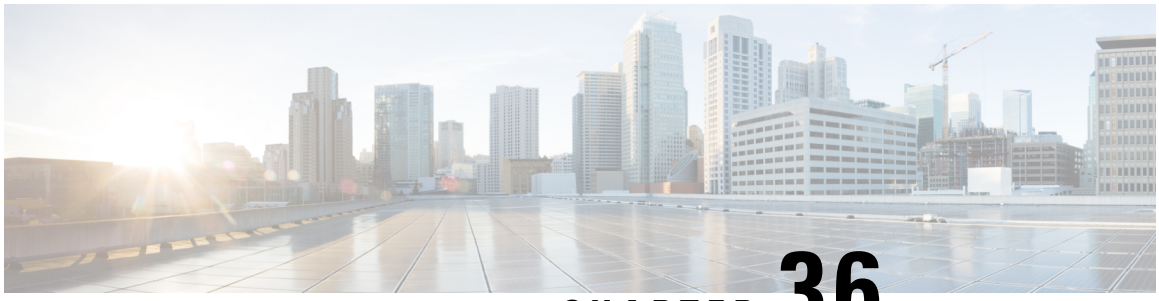
Technical Assistance

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 SIP ALG Hardening for NAT and Firewall

Table 46: Feature Information for SIP ALG Hardening for NAT and Firewall

Feature Name	Releases	Feature Information
SIP ALG Hardening for NAT and Firewall	Cisco IOS XE Release 3.8S	The SIP ALG Hardening for NAT and Firewall feature provides better memory management and RFC compliance over the existing SIP ALG support for NAT and firewall.



CHAPTER 36

SIP ALG Resilience to DoS Attacks

The SIP ALG Resilience to DoS Attacks feature provides protection against Session Initiation Protocol (SIP) application layer gateway (ALG) denial of service (DoS) attacks. This feature supports a configurable lock limit, a dynamic blacklist, and configurable timers to prevent DoS attacks.

This module explains the feature and how to configure DoS prevention for the SIP application layer gateway (ALG). Network Address Translation and zone-based policy firewalls support this 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.

Information About SIP ALG Resilience to DoS Attacks

SIP ALG Resilience to DoS Attacks Overview

The SIP ALG Resilience to DoS Attacks feature provides protection against denial of service (DoS) attacks to the Session Initiation Protocol (SIP) application layer gateway (ALG). This feature supports a configurable lock limit, a dynamic blacklist, and configurable timers to prevent DoS attacks. This feature is supported by Network Address Translation (NAT) and zone-based policy firewalls.

SIP is an application-level signaling protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP DoS attacks are a major threat to networks.

The following are types of SIP DoS attacks:

- SIP register flooding: A registration flood occurs when many VoIP devices try to simultaneously register to a network. If the volume of registration messages exceeds the device capability, some messages are lost. These devices then attempt to register again, adding more congestion. Because of the network congestion, users may be unable to access the network for some time.
- SIP INVITE flooding: An INVITE flood occurs when many INVITE messages are sent to servers that cannot support all these messages. If the attack rate is very high, the memory of the server is exhausted.
- SIP broken authentication and session attack: This attack occurs when an attacker presumes the identity of a valid user, using digest authentication. When the authentication server tries to verify the identity of the attacker, the verification is ignored and the attacker starts a new request with another session identity. These attacks consume the memory of the server.

SIP ALG Dynamic Blacklist

One of the common methods of denial of service (DoS) attacks involves saturating the target network with external communication requests making the network unable to respond to legitimate traffic. To solve this issue, the SIP ALG Resilience to DoS Attacks feature uses configurable blacklists. A blacklist is a list of entities that are denied a particular privilege, service, or access. Dynamic blacklists are disabled by default. When requests to a destination address exceed a predefined trigger criteria in the configured blacklist, the Session Initiation Protocol (SIP) application layer gateway (ALG) will drop these packets.

The following abnormal SIP session patterns are monitored by dynamic blacklists:

- In the configured period of time if a source sends multiple requests to a destination and receives non-2xx (as per RFC 3261, any response with a status code between 200 and 299 is a "2xx response") final responses from the destination.
- In the configured period of time if a source sends multiple requests to a destination and does not receive any response from the destination.

SIP ALG Lock Limit

Both Network Address Translation (NAT) and the firewall use the Session Initiation Protocol (SIP) application layer gateway (ALG) to parse SIP messages and create sessions through tokens. To maintain session states, the SIP ALG uses a per call data structure and Layer 7 data to store call-related information that is allocated when a session is initiated and freed when a session is released. If the SIP ALG does not receive a message that indicates that the call has ended, network resources are held for the call.

Because Layer 7 data is shared between threads, a lock is required to access the data. During denial of service (DoS) and distributed DoS attacks, many threads wait to get the same lock, resulting in heavy CPU usage, which makes the system unstable. To prevent the system from becoming unstable, a limit is added to restrict the number of threads that can wait for a lock. SIP sessions are established by request/response mode. When there are too many concurrent SIP messages for one SIP call, packets that exceed the lock limit are dropped.

SIP ALG Timers

To exhaust resources on Session Initiation Protocol (SIP) servers, some denial of service (DoS) attacks do not indicate the end of SIP calls. To prevent these types of DoS attacks, a protection timer is added.

The SIP ALG Resilience to DoS Attacks feature uses the following timers:

- Call-duration timer that controls the maximum length of an answered SIP call.
- Call-proceeding timer that controls the maximum length of an unanswered SIP call.

When the configured maximum time is reached, the SIP application layer gateway (ALG) releases resources for this call, and future messages related to this call may not be properly parsed by the SIP ALG.

How to Configure SIP ALG Resilience to DoS Attacks

Configuring SIP ALG Resilience to DoS Attacks

You can configure the prevention of denial of service (DoS) parameters for the Session Initiation Protocol (SIP) application layer gateway (ALG) that is used by Network Address Translation (NAT) and the zone-based policy firewall.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **alg sip processor session max-backlog** *concurrent-processor-usage*
4. **alg sip processor global max-backlog** *concurrent-processor-usage*
5. **alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **destination** *ip-address*
6. **alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **block-time** *block-time* [**destination** *ip-address*]
7. **alg sip timer call-proceeding-timeout** *time*
8. **alg sip timer max-call-duration** *seconds*
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	**alg sip processor session max-backlog** *concurrent-processor-usage* **Example:** `Device(config)# alg sip processor session max-backlog 5`	Sets a per session limit for the number of backlog messages waiting for shared resources.

	Command or Action	Purpose
Step 4	**alg sip processor global max-backlog** *concurrent-processor-usage* **Example:** `Device(config)# alg sip processor global max-backlog 5`	Sets the maximum number of backlog messages waiting for shared resources for all SIP sessions.
Step 5	**alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **destination** *ip-address* **Example:** `Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 destination 10.1.1.1`	Configures dynamic SIP ALG blacklist criteria for the specified destination IP address.
Step 6	**alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **block-time** *block-time* [**destination** *ip-address*] **Example:** `Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 block-time 30`	Configures the time period, in seconds, when packets from a source are blocked if the configured limit is exceeded.
Step 7	**alg sip timer call-proceeding-timeout** *time* **Example:** `Device(config)# alg sip timer call-proceeding-timeout 35`	Sets the maximum time interval, in seconds, to end SIP calls that do not receive a response.
Step 8	**alg sip timer max-call-duration** *seconds* **Example:** `Device(config)# alg sip timer max-call-duration 90`	Sets the maximum call duration, in seconds, for a successful SIP call.
Step 9	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying SIP ALG Resilience to DoS Attacks

Use the following commands to troubleshoot the feature.

SUMMARY STEPS

1. **enable**
2. **show alg sip**
3. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip**
4. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dbl**
5. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dblcfg**
6. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip processor**
7. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip timer**

8. **debug alg** {**all** | **info** | **trace** | **warn**}

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show alg sip**

Displays all Session Initiation Protocol (SIP) application layer gateway (ALG) information.

Example:

```
Device# show alg sip

sip timer configuration
   Type                      Seconds
   max-call-duration         380
   call-proceeding-timeout   620

sip processor configuration
   Type           Backlog number
   session         14
   global          189

sip blacklist configuration
   dst-addr         trig-period(ms)    trig-size     block-time(sec)
   10.0.0.0             60                 30              2000
   10.1.1.1             20                 30              30
   192.0.2.115          1000               5               30
   198.51.100.34        20                 30              388
```

Step 3 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip**

Displays SIP ALG-specific statistics information in the Cisco Quantum Flow Processor (QFP).

Example:

```
Device# show platform hardware qfp active feature alg statistics sip

Events
...
Cr dbl entry:                10   Del dbl entry:                 10
Cr dbl cfg entry:             8   Del dbl cfg entry:              4
start dbl trig tmr:          10   restart dbl trig tmr:        1014
stop dbl trig tmr:           10   dbl trig timeout:            1014
start dbl blk tmr:            0   restart dbl blk tmr:            0
stop dbl blk tmr:             0   dbl blk tmr timeout:            0
start dbl idle tmr:          10   restart dbl idle tmr:         361
stop dbl idle tmr:            1   dbl idle tmr timeout:           9

DoS Errors
Dbl Retmem Failed:            0    Dbl Malloc Failed:             0
DblCfg Retm Failed:           0    DblCfg Malloc Failed:          0
Session wlock ovflw:          0    Global wlock ovflw:            0
Blacklisted:                561
```

Step 4 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dbl**

Displays brief information about all SIP blacklist data.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip dbl

SIP dbl pool used chunk entries number: 1

 entry_id          src_addr         dst_addr         remaining_time(sec)
 a4a051e0a4a1ebd   10.74.30.189     10.74.5.30               25
```

Step 5 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dblcfg**

Displays all SIP blacklist settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip dblcfg

SIP dbl cfg pool used chunk entries number: 4
 dst_addr        trig_period(ms)    trig_size    block_time(sec)
 10.1.1.1              20              30             30
 10.74.5.30            1000             5             30
 192.0.2.2             60              30           2000
 198.51.100.115        20              30            388
```

Step 6 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip processor**

Displays SIP processor settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip processor

Session:     14        Global:    189

Current global wlock count:      0
```

Step 7 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip timer**

Displays SIP timer settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip timer

call-proceeding:    620      call-duration:    380
```

Step 8 **debug alg** {**all** | **info** | **trace** | **warn**}

Example:

```
Device# debug alg warn
```

Enables the logging of ALG warning messages.

Configuration Examples for SIP ALG Resilience to DoS Attacks

Example: Configuring SIP ALG Resilience to DoS Attacks

```
Device# configure terminal
Device(config)# alg sip processor session max-backlog 5
Device(config)# alg sip processor global max-backlog 5
Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 destination 10.1.1.1
Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 block-time 30
Device(config)# alg sip timer call-proceeding-timeout 35
Device(config)# alg sip timer max-call-duration 90
Device(config)# end
```

Additional References for SIP ALG Resilience to DoS Attacks

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
NAT commands	IP Addressing Services Command References

Standards and RFCs

Standard/RFC	Title
RFC 4028	*Session Timers in the Session Initiation Protocol (SIP)*

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

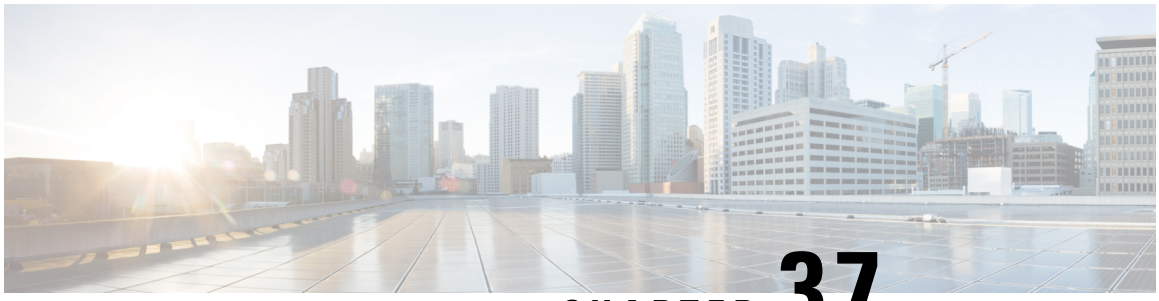
Technical Assistance

Description	Link
The Cisco Support 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 SIP ALG Resilience to DoS Attacks

Table 47: Feature Information for SIP ALG Resilience to DoS Attacks

Feature Name	Releases	Feature Information
SIP ALG Resilience to DoS Attacks	Cisco IOS XE Release 3.11S	The SIP ALG Resilience to DoS Attacks feature provides protection against Session Initiation Protocol (SIP) denial of service (DoS) attacks. This feature supports a configurable lock limit, a dynamic blacklist, and configurable timers to prevent DoS attacks. Network Address Translation (NAT) and zone-based policy firewalls support this feature. In Cisco IOS XE Release 3.11S, the SIP ALG Resilience to DoS Attacks feature is implemented on Cisco ASR 1000 Series Aggregation Services Routers, Cisco Cloud Services Routers 1000V Series, and Cisco 4400 Series Integrated Services Routers. The following commands were introduced or modified: **alg sip processor**, **alg sip blacklist**, **alg sip timer**, **show alg sip**, **debug alg**, **debug platform software alg configuration all**, **set platform software trace forwarding-manager alg**, and **show platform hardware qfp feature alg statistics sip**.



CHAPTER 37

Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

The Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support feature supports the following functionalities for Application Layer Gateway (ALG), and Application Inspection and Control (AIC):

- Packet tracing
- Conditional debugging
- Debug logs

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 Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

Packet Tracing

Packet tracing provides the ability to generate Control Plane Policing (CPP) statistics for a specified packet flow, with minimal effect on router throughput. It also traces the path of each packet in the flow, which helps in determining the input interface, features used, and the output path.

Application layer gateway (ALG) generates statistics and keeps a log of the path along which the packets travel.

Conditional Debugging

In a typical Application layer gateway (ALG)-enabled scenario where certain connections from the source address or destination address fail, debugging displays a list of messages for all the traffic that passes through the ALG. Enabling conditional debugging ensures that debug messages related to specified connections are displayed on the console. Prior to the introduction of this feature, debugging used to display many messages for all traffic that passes through the ALG.

Debug Logs

The following severity levels have been added:

1. Error: Error and firewall packet drop conditions.

   Examples:

   - Unable to send a packet
   - ALG error condition

2. Warning: Warning debug messages.
3. Info: Information about an event.

   Examples:

   - Packet drop due to policy configuration, malformed packets, or hardcoded limit and threshold
   - State machine transition
   - ALG check status
   - Packet pass and drop status

4. Verbose: All log messages.

   Examples:

   - Data structures

- Event details

Note Both the ALG-AIC functional debug flag and the severity level must be set. If only the severity level is set and the ALG-AIC functional debug flag is not set, the debug log will not be enabled. If only the ALG-AIC functional debug flag is set, the Info level, which is the default severity level, is logged.

Additional References for Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

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

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 ALG and AIC Conditional Debugging and Packet Tracing 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 48: Feature Information for Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

Feature Name	Releases	Feature Information
Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support	Cisco IOS XE 3.13S	The Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support feature supports the following functionalities: • Packet tracing • Conditional debugging • Debug logs