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.
- 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 AppNav I/O modules. Cisco AppNav 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 are used only for classification; the firewall does not display the packet count that matches the configured ACLs. Perfilter statistics is available in zone-based firewalls from Cisco IOS XE Release 3.13S and later releases.
- Bridge domain interfaces do not support zone-based firewall inspection, including all Layer 4 and Layer 7 inspection.
- 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 **pass** action and **inspect** action are configured. In Cisco IOS XE Release 3.15S and later releases, the zone-based firewall supports zone mismatch traffic. Based on your configuration, you can configure the firewall to drop zone mismatch traffic flow. In releases prior to Cisco IOS XE Release 3.15s, the zone mismatch traffic is by default inspected.

- The zone-based firewall is not supported along with dynamic interfaces. These interfaces are created or deleted dynamically when traffic is tunneled into tunnels such as IPsec or VPN secure tunnels.
- 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. 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:

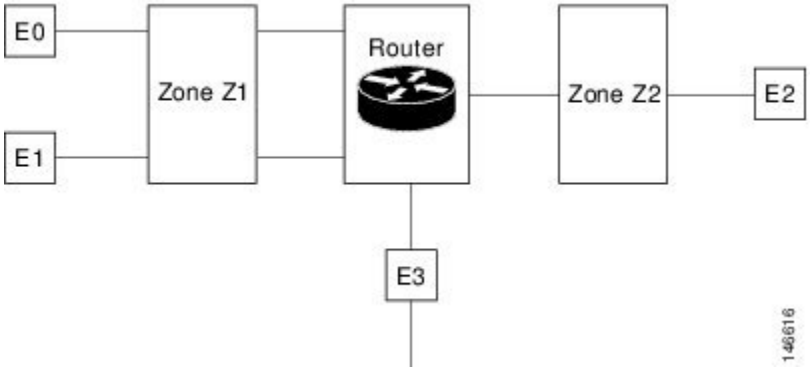
- Traffic from a zone interface to a nonzone interface or from a nonzone interface to a zone interface is always dropped; unless default zones are enabled (default zone is a nonzone interface).
- Traffic between two zone interfaces is inspected if there is a zone pair relationship for each zone and if 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 or drop the traffic between two interfaces in the same zone.
- An interface can be a member of only one security zone.

- When an interface is a member of a security zone, all traffic to and from that interface is blocked unless you configure an explicit interzone policy on a zone pair involving that zone.
- Traffic cannot flow between an interface that is a member of a security zone and an interface that is not a member of a security zone because a policy can be applied only between two zones.
- For traffic to flow among all interfaces in a device, 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.

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. For example, **inspect**, **drop**, and **pass** are actions.

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.

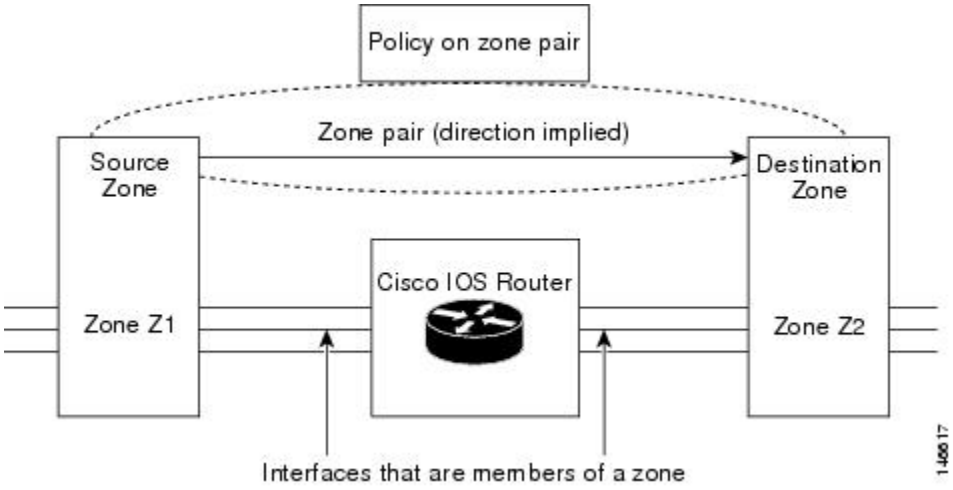
You can select the default or self zone as either the source or the destination zone. The self zone is a system-defined zone which does not have any interfaces as members. A zone pair that includes the self zone, along with the associated policy, applies to traffic directed to the device or traffic generated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

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

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

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

Figure 2: Zone Pairs



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

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

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

A zone-based firewall behaves differently when handling intermittent Internet Control Message Protocol (ICMP) responses generated within a zone because of the traffic flowing between in-zones and out-zones.

In a configuration where an explicit policy is configured for the self zone to go out of its zone and for the traffic moving between the in-zone and out-zone, if any intermittent ICMP responses are generated, then the zone-based firewall looks for an explicit permit rule for the ICMP in the self zone to go out of its zone. An explicit inspect rule for the ICMP for the self zone to go out-zone may not help because there is no session associated with the intermittent ICMP responses.

Zones and Inspection

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

will create a connection in the firewall table and be subject to state checking. Traffic with the pass action will bypass the zone firewall completely, not creating any sessions.

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

Zones and ACLs

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

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

Class Maps and Policy Maps for Zone-Based Policy Firewalls

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

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

Layer 3 and Layer 4 Class Maps and Policy Maps

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

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

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

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

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
- ICMP
- Lightweight Directory Access Protocol (LDAP)
- LDAP over Transport Layer Security/Secure Socket Layer (LDAPS)
- Real-time Streaming Protocol (RTSP)
- Session Initiation Protocol (SIP)
- SCCP (Skinny Client Control Protocol)

- TCP
- TFTP
- UDP

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 particular router network interface. 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.
- If a class map is specified, the class map performs either an AND (match-all) or an OR (match-any) operation on the ACL variables.
- 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.

Parameter Maps

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

There are two types of parameter maps:

- Inspect parameter map

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

- Protocol-specific parameter map

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

Firewall and Network Address Translation

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

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

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

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

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

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

WAAS Support for the Cisco Firewall

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

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

WAAS has an automatic discovery mechanism that uses TCP options during the initial three-way handshake to identify WAE devices transparently. After automatic discovery, optimized traffic flows (paths) experience

a change in the TCP sequence number to allow endpoints to distinguish between optimized and nonoptimized traffic flows.

Note Paths are synonymous with connections.

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

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

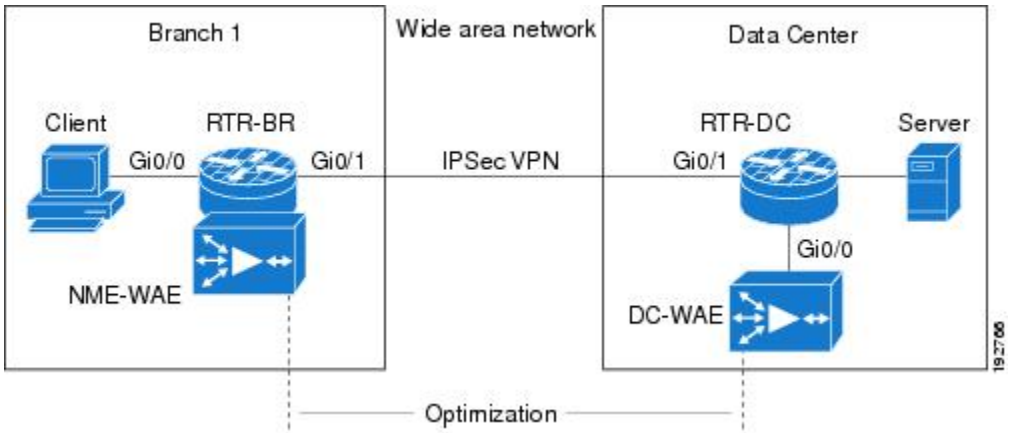
Note Stateful Layer 7 inspection on the client side can also be performed on nonoptimized traffic.

WAAS Traffic Flow Optimization Deployment Scenarios

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

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

Figure 3: End-to-End WAAS Optimization Path

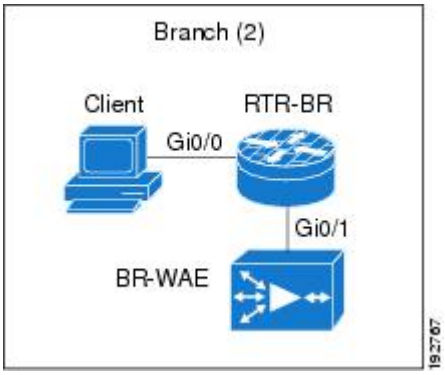


WAAS Branch Deployment with an Off-Path Device

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.

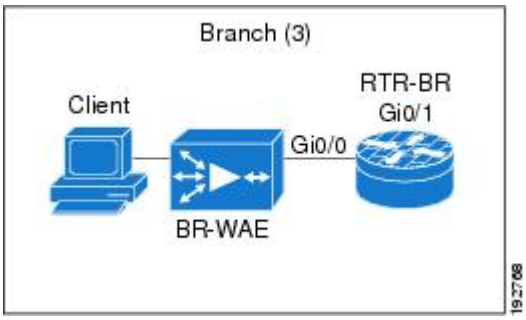
Figure 4: WAAS Off-Path Branch Deployment



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:** `Device(config-cmap)# match protocol http`	Configures the match criterion for a class map on the basis of a specified protocol. • Only Cisco stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps. • **signature—**Signature-based classification for peer-to-peer packets is enabled.

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

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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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.

Use this process to complete the following tasks:

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

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **description** *line-of-description*
5. **exit**
6. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
7. **description** *line-of-description*
8. **exit**
9. **interface** *type number*
10. **zone-member security** *zone-name*
11. **exit**
12. **zone-pair security** *zone-pair-name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
13. **service-policy type inspect** *policy-map-name*
14. **platform inspect match-statistics per-filter**
15. **end**

DETAILED STEPS

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

	Command or Action	Purpose
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	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 7	**description** *line-of-description* **Example:** `Device(config-sec-zone-pair)# description accounting network`	(Optional) Describes the zone pair.
Step 8	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Configures an interface and enters interface configuration mode.
Step 10	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you should apply a policy. If the policy permits traffic, traffic can flow through that interface.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**zone-pair security** *zone-pair-name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode.
Step 13	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p2`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 14	**platform inspect match-statistics per-filter** **Example:** `Device(config-sec-zone-pair)# platform inspect match-statistics per-filter`	Enables zone-based firewall per-filter statistics. **Note** To enable per-filter statistics on the device, do the following: • RELOAD the device. • OR Remove all the service-policies and re-apply the changes to the statistics. To activate the **platform inspect match-statistics per-filter** command, re-apply all service-policies.
Step 15	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to privileged EXEC mode.

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 dropped packet logging.
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.
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
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.

In Cisco IOS XE software, WAAS support is enabled by default and WAAS processing is discovered.

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. **parameter-map type inspect global**
6. **waas enable**
7. **log dropped-packets enable**
8. **max-incomplete low**
9. **max-incomplete high**
10. **ip inspect waas enable**
11. **class-map type inspect** *class-name*
12. **match protocol** *protocol-name* [**signature**]
13. **exit**
14. **policy-map type inspect** *policy-map-name*
15. **class class-default**
16. **class-map type inspect** *class-name*
17. **inspect**
18. **exit**
19. **exit**
20. **zone security** *zone-name*
21. **description** *line-of-description*
22. **exit**
23. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
24. **description** *line-of-description*
25. **exit**
26. **interface** *type number*
27. **description** *line-of-description*
28. **zone-member security** *zone-name*
29. **ip address** *ip-address*
30. **ip wccp** *service-id* {**group-listen** | **redirect** {**in** | **out**}}
31. **exit**
32. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**} **destination** [**self** | *destination-zone-name*]
33. **service-policy type inspect** *policy-map-name*
34. **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	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	
Step 6	**waas enable** **Example:** `Device(config-profile)# waas enable`	
Step 7	**log dropped-packets enable** **Example:** `Device(config-profile)# log dropped-packets enable`	
Step 8	**max-incomplete low** **Example:** `Device(config)# max-incomplete low 18000`	
Step 9	**max-incomplete high** **Example:** `Device(config)# max-incomplete high 20000`	

	Command or Action	Purpose
Step 10	**ip inspect waas enable** **Example:** l `Device(config)# ip inspect waas enable`	Enables firewall inspection so that Cisco Wide Area Application Service (WAAS) optimization can be discovered. **Note**
Step 11	**class-map type inspect** *class-name* **Example:** `Device(config)# class-map type inspect most-traffic`	Creates an inspect type class map for the traffic class and enters class-map configuration mode. **Note** The **class-map type inspect most-traffic** command is hidden.
Step 12	**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 13	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
Step 14	**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 15	**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 16	**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 17	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables Cisco stateful packet inspection.
Step 18	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.

	Command or Action	Purpose
Step 19	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode and returns to global configuration mode.
Step 20	**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 21	**description** *line-of-description* **Example:** `Device(config-sec-zone)# description Internet Traffic`	(Optional) Describes the zone.
Step 22	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 23	**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 24	**description** *line-of-description* **Example:** `Device(config-sec-zone)# description accounting network`	(Optional) Describes the zone pair.
Step 25	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 26	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters interface configuration mode.
Step 27	**description** *line-of-description* **Example:** `Device(config-if)# description zone interface`	(Optional) Describes an interface.
Step 28	**zone-member security** *zone-name*	Assigns an interface to a specified security zone.

	Command or Action	Purpose
	Example: `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 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 29	**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.
Step 30	**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 31	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 32	**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 33	**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 34	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to privileged EXEC mode.

Configuring LDAP-Enabled Firewalls

Lightweight Directory Access Protocol (LDAP) is an application protocol that is used for querying and updating information stored on directory servers. The LDAP-Enabled Firewall feature enables Cisco firewalls to support Layer 4 LDAP inspection by default.

You can configure an LDAP-enabled firewall in interface configuration mode or in global configuration mode. Before you configure an LDAP-enabled firewall in interface configuration mode, you must configure a zone by using the **zone security** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **class-map type inspect** [**match-all** | **match-any**] *class-map-name*
8. **match protocol** *protocol-name*
9. **exit**
10. **policy-map type inspect match-any** *policy-map-name*
11. **class type inspect** *class-name*
12. **inspect**
13. **class class-default**
14. **exit**
15. **exit**
16. **zone-pair security** *zone-pair-name* **source** {*source-zone-name* | **self default**} **destination** {*destination-zone-name* | **self default**}
17. **service-policy type inspect** *policy-map-name*
18. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security private`	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 internet`	Exits security zone configuration mode and returns to global configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and global configuration mode.
Step 7	**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 for the traffic class and returns to class-map configuration mode.
Step 8	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol ldap`	Configures the match criteria for a class map based on the specified protocol.
Step 9	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and returns to global configuration mode.
Step 10	**policy-map type inspect match-any** *policy-map-name* **Example:** `Device(config)# policy-map type inspect match-any private-internet-policy`	Creates a Layer 3 or Layer 4 inspect type policy map and enters policy map configuration mode.
Step 11	**class type inspect** *class-name* **Example:** `Device(config-pmap)# class type inspect internet-traffic-class`	Specifies the firewall traffic (class) map on which an action is to be performed and enters policy-map class configuration mode..
Step 12	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.

	Command or Action	Purpose
Step 13	**class class-default** **Example:** `Device(config-pmap-c)# class class-default`	Specifies the matching of the system default class. • If the system default class is not specified, then unclassified packets are matched.
Step 14	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy map configuration mode.
Step 15	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy map configuration mode and returns to global configuration mode.
Step 16	**zone-pair security** *zone-pair-name* **source** {*source-zone-name* | **self default**} **destination** {*destination-zone-name* | **self default**} **Example:** `Device(config)# zone-pair security private-internet source private destination internet`	Creates a zone pair and enters security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 17	**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 18	**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
 WAAS enable
 log dropped-packets enable
 max-incomplete low 18000
 max-incomplete high 20000
!
class-map type inspect match-any most-traffic
 match protocol icmp
 match protocol ftp
 match protocol tcp
 match protocol udp
!
policy-map type inspect p1
 class type inspect most-traffic
  inspect
!
 class class-default
  drop
!
zone security in
!
zone security out
!
zone security waas
!
zone-pair security in-out source in destination out
 service-policy type inspect p1
!
zone-pair security out-in source out destination in
 service-policy type inspect p1
!
zone-pair security waas-out source waas destination out
 service-policy type inspect p1
!
zone-pair security in-waas source in destination waas
 service-policy type inspect p1
!
interface GigabitEthernet0/0
 description WAN Connection
 no ip dhcp client request tftp-server-address
 no ip dhcp client request router
```

```
 ip address dhcp
 ip wccp 62 redirect in
 ip wccp 61 redirect out
 ip flow ingress
 ip nat outside
 ip virtual-reassembly in
 ip virtual-reassembly out
 zone-member security out
 load-interval 30
 delay 30
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 description Clients
 ip address 172.25.50.1 255.255.255.0
 ip pim sparse-mode
 ip nat inside
 ip virtual-reassembly in
 zone-member security in
 ip igmp version 3
 delay 30
 duplex auto
 speed auto
!
interface Vlan1
 description WAAS Interface
 ip address 172.25.60.1 255.255.255.0
 ip wccp redirect exclude in
 ip nat inside
 ip virtual-reassembly in
 zone-member security waas
 load-interval 30
!
```

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

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

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

Note The new configuration, depending on your release, places an integrated service engine in its own zone and need not be part of any zone pair. The zone pairs are configured between zone-hr (zone-out) and zone-eng (zone-output).

```
interface Integrated-Service-Engine 1/0
 ip address 10.70.100.1 255.255.255.252
 ip wccp redirect exclude in
 zone-member security z-waas
```

Example: Configuring LDAP-Enabled Firewalls

Interface Configuration

```
interface GigabitEthernet 0/1/5
 ip address 192.168.0.1 255.255.255.0
 zone-member security private
 no shutdown
!
interface GigabitEthernet 0/1/6
 ip address 192.168.1.1 255.255.255.0
 zone-member security internet
 no shutdown
!
```

Global Firewall Configuration

```
zone security private
!
zone security internet
!
class-map type inspect match-any internet-traffic-class
 match protocol ldap
 match protocol ldaps
 match protocol ldap-admin
!
policy-map type inspect private-internet-policy
 class type inspect internet-traffic-class
  inspect
!
 class class-default
!
zone-pair security private-internet source private destination internet
 service-policy type inspect private-internet-policy
!
```

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 2.4	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.
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 XE 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.

Feature Name	Releases	Feature Information
Zone-Based Firewall—Default Zone	Cisco IOS XE 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 a default zone. The following commands were introduced by this feature: **zone pair security**, **zone security**.
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.