Cisco IOS XR Carrier Grade NAT Configuration Guide for the Cisco CRS Router

Cisco IOS XR Software Release 4.0.0

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

Customer Order Number: OL-23341-01

Cisco IOS XR Carrier Grade NAT Configuration Guide for the Cisco CRS Router

CONTENTS

Index

Preface

The *Cisco IOS XR Carrier Grade NAT Configuration Guide for the Cisco CRS Router* preface contains the following sections:

Changes to This Document

Table 1 lists the technical changes made to this document since it was first printed.

Table 1 Changes to This Document

Revision	Date	Change Summary
OL-23341-01	September 2010	Initial release of this document.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

Implementing the Carrier Grade NAT on Cisco IOS XR Software

This module describes how to implement the Carrier Grade NAT (CGN) on Cisco IOS XR software.

Contents

Prerequisites for Implementing the Carrier Grade NAT

The following prerequisites are required to implement Carrier Grade NAT:

- You must be running *Cisco IOS XR software Release 3.9.1* or above.
- You must have installed the CGN service package or pie **hfr-cgn-p.pie-x.x.x** (where x.x.x specifies the release number of Cisco IOS XR software).
- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command.
- In case of intra chassis redundancy, enable CGSE data and control path monitoring in configuration mode, where R/S/CPU0 is the CGSE Location -
  - service-plim-ha location is R/S/CPU0 datapath-test
  - service-plim-ha location is R/S/CPU0 core-to-core-test
  - service-plim-ha location is R/S/CPU0 pci-test
  - service-plim-ha location is R/S/CPU0 coredump-extraction
  - service-plim-ha location 0/0/CPU0 linux-timeout 500
  - service-plim-ha location 0/0/CPU0 msc-timeout 500

Note All the error conditions result in card reload that triggers switchover to standby CGSE. The option of revertive switchover (that is disabled by default) and forced switchover is also available and can be used if required. Contact Cisco Technical Support with **show tech-support cgn** information.

- In case of standalone CGSEs (without intra chassis redundancy), enable CGSE data and control path monitoring in configuration mode, where R/S/CPU0 is the CGSE Location with auto reload disabled and
  - service-plim-ha location R/S/CPU0 datapath-test
  - service-plim-ha location R/S/CPU0 core-to-core-test
  - service-plim-ha location R/S/CPU0 pci-test
  - service-plim-ha location R/S/CPU0 coredump-extraction
  - service-plim-ha location 0/0/CPU0 linux-timeout 500
  - service-plim-ha location 0/0/CPU0 msc-timeout 500
  - (admin-config) hw-module reset auto disable location R/S/CPU0

Note All the error conditions result in a syslog message. On observation of Heartbeat failures or any HA test failure messages, contact Cisco Technical Support with **show tech-support cgn** information.

Note If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Carrier Grade NAT Overview and Benefits

To implement the Carrier Grade NAT, you should understand the following concepts:

Carrier Grade NAT Overview

Carrier Grade Network Address Translation (CGN) is a large scale NAT that is capable of providing private IPv4 to public IPv4 address translation in the order of millions of translations to support a large number of subscribers, and at least 10 Gbps full-duplex bandwidth throughput.

CGN is a workable solution to the IPv4 address completion problem, and offers a way for service provider subscribers and content providers to implement a seamless transition to IPv6. CGN employs network address and port translation (NAPT) methods to aggregate many private IP addresses into fewer public IPv4 addresses. For example, a single public IPv4 address with a pool of 32 K port numbers supports 320 individual private IP subscribers assuming each subscriber requires 100 ports. For example, each TCP connection needs one port number.

A CGN requires IPv6 to assist with the transition from IPv4 to IPv6.

Benefits of Carrier Grade NAT

CGN offers these benefits:

- Enables service providers to execute orderly transitions to IPv6 through mixed IPv4 and IPv6 networks.
- Provides address family translation but not limited to just translation within one address family.
- Delivers a comprehensive solution suite for IP address management and IPv6 transition.

IPv4 Address Shortage

A fixed-size resource such as the 32-bit public IPv4 address space will run out in a few years. Therefore, the IPv4 address shortage presents a significant and major challenge to all service providers who depend on large blocks of public or private IPv4 addresses for provisioning and managing their customers.

Service providers cannot easily allocate sufficient public IPv4 address space to support new customers that need to access the public IPv4 Internet.

NAT and NAPT Overview

A Network Address Translation (NAT) box is positioned between private and public IP networks that are addressed with non-global private addresses and a public IP addresses respectively. A NAT performs the task of mapping one or many private (or internal) IP addresses into one public IP address by employing both network address and port translation (NAPT) techniques. The mappings, otherwise referred to as bindings, are typically created when a private IPv4 host located behind the NAT initiates a connection (for example, TCP SYN) with a public IPv4 host. The NAT intercepts the packet to perform these functions:

- Rewrites the private IP host source address and port values with its own IP source address and port values
- Stores the private-to-public binding information in a table and sends the packet. When the public IP host returns a packet, it is addressed to the NAT. The stored binding information is used to replace the IP destination address and port values with the private IP host address and port values.

Traditionally, NAT boxes are deployed in the residential home gateway (HGW) to translate multiple private IP addresses. The NAT boxes are configured on multiple devices inside the home to a single public IP address, which are configured and provisioned on the HGW by the service provider. In enterprise scenarios, you can use the NAT functions combined with the firewall to offer security protection for corporate resources and allow for provider-independent IPv4 addresses. NATs have made it easier for private IP home networks to flourish independently from service provider IP address provisioning. Enterprises can permanently employ private IP addressing for Intranet connectivity while relying on a few NAT boxes, and public IPv4 addresses for external public Internet connectivity. NAT boxes in conjunction with classic methods such as Classless Inter-Domain Routing (CIDR) have slowed public IPv4 address consumption.

Network Address and Port Mapping

Network address and port mapping can be reused to map new sessions to external endpoints after establishing a first mapping between an internal address and port to an external address. These NAT mapping definitions are defined from RFC 4787:

- **Endpoint-independent mapping**—Reuses the port mapping for subsequent packets that are sent from the same internal IP address and port to any external IP address and port.
- **Address-dependent mapping**—Reuses the port mapping for subsequent packets that are sent from the same internal IP address and port to the same external IP address, regardless of the external port.

Translation Filtering

RFC 4787 provides translation filtering behaviors for NATs. These options are used by NAT to filter packets originating from specific external endpoints:

- **Endpoint-independent filtering**—Filters out only packets that are not destined to the internal address and port regardless of the external IP address and port source.
- **Address-dependent filtering**—Filters out packets that are not destined to the internal address. In addition, NAT filters out packets that are destined for the internal endpoint.
- **Address and port-dependent filtering**—Filters out packets that are not destined to the internal address. In addition, NAT filets out packets that are destined for the internal endpoint if the packets were not sent previously.

Information About Implementing Carrier Grade NAT

These sections provide the information about implementation of NAT using ICMP and TCP:

Implementing NAT with ICMP

This section explains how the Network Address Translation (NAT) devices work in conjunction with Internet Control Message Protocol (ICMP).

The implementations of NAT varies in terms of how they handle different traffic.

ICMP Query Session Timeout

RFC 5508 provides ICMP Query Session timeouts. A mapping timeout is maintained by NATs for ICMP queries that traverse them. The ICMP Query Session timeout is the period during which a mapping will stay active without packets traversing the NATs. The timeouts can be set as either *Maximum Round Trip Time* (Maximum RTT) or *Maximum Segment Lifetime* (MSL). For the purpose of constraining the maximum RTT, the Maximum Segment Lifetime (MSL) is considered a guideline to set packet lifetime.

If the ICMP NAT session timeout is set to a very large duration (240 seconds) it can tie up precious NAT resources such as Query mappings and NAT Sessions for the whole duration. Also, if the timeout is set to very low it can result in premature freeing of NAT resources and applications failing to complete gracefully. The ICMP Query session timeout needs to be a balance between the two extremes. A 60-second timeout is a balance between the two extremes.

Implementing NAT with TCP

This section explains the various NAT behaviors that are applicable to TCP connection initiation. The detailed NAT with TCP functionality is defined in RFC 5382.

Address and Port Mapping Behavior

A NAT translates packets for each TCP connection using the mapping. A mapping is dynamically allocated for connections initiated from the internal side, and potentially reused for certain connections later.

Internally Initiated Connections

A TCP connection is initiated by internal endpoints through a NAT by sending SYN packet. All the external IP address and port used for translation for that connection are defined in the mapping.

Generally for the client-server applications where an internal client initiates the connection to an external server, to translate the outbound SYN, the resulting inbound SYN-ACK response mapping is used, the subsequent outbound ACK, and other packets for the connection.

The 3-way handshake corresponds to method of connection initiation.

Externally Initiated Connections

For the first connection that is initiated by an internal endpoint NAT allocates the mapping. For some situations, the NAT policy may allow reusing of this mapping for connection initiated from the external side to the internal endpoint.

Double NAT 444

The Double NAT 444 solution offers the fastest and simplest way to address the IPv4 depletion problem without requiring an upgrade to IPv6 anywhere in the network. Service providers can continue offering new IPv4 customers access to the public IPv4 Internet by using private IPv4 address blocks, if the service provider is large enough; However, they need to have an overlapping RFC 1918 address space, which forces the service provider to partition their network management systems and creates complexity with access control lists (ACL).

Double NAT 444 uses the edge NAT and CGN to hold the translation state for each session. For example, both NATs must hold 100 entries in their respective translation tables if all the hosts in the residence of a subscriber have 100 connections to hosts on the Internet). There is no easy way for a private IPv4 host to communicate with the CGN to learn its public IP address and port information or to configure a static incoming port forwarding.

Address Family Translation

The IPv6-only to IPv4-only protocol is referred to as address family translation (AFT). The AFT translates the IP address from one address family into another address family. For example, IPv6 to IPv4 translation is called NAT 64 or IPv4 to IPv6 translation is called NAT 46.

Policy Functions

- Application Level Gateway, page 6
- TCP Maximum Segment Size Adjustment, page 6
- Static Port Forwarding, page 6

Application Level Gateway

The application level gateway (ALG) deals with the applications that are embedded in the IP address payload. Therefore, the active FTP ALG is supported.

CGN supports both passive and active FTP. FTP clients are supported with inside (private) address and servers with outside (public) addresses. Passive FTP is provided by the basic NAT function. Active FTP is used with the ALG.

TCP Maximum Segment Size Adjustment

When a host initiates a TCP session with a server, the host negotiates the IP segment size by using the maximum segment size (MSS) option. The value of the MSS option is determined by the maximum transmission unit (MTU) that is configured on the host.

Static Port Forwarding

Static port forwarding configures a fixed, private (internal) IP address and port that are associated with a particular subscriber while CGN allocates a free public IP address and port. Therefore, the inside IP address and port are associated to a free outside IP address and port.

External Logging

External logging configures the export and logging of the NAT table entries, private bindings that are associated with a particular global IP port address, and to use Netflow to export the NAT table entries.

Implementing Carrier Grade NAT on Cisco IOS XR Software

The following configuration tasks are required to implement CGN on Cisco IOS XR software:

- Getting Started with the Carrier Grade NAT, page 7
- Configuring the Service Type Keyword Definition, page 13
- Configuring the Policy Functions for the Carrier Grade NAT, page 16
- Configuring the Export and Logging for the Network Address Translation Table Entries, page 29

Getting Started with the Carrier Grade NAT

Perform these tasks to get started with the CGN configuration tasks.

- Configuring the Service Role, page 7
- Configuring the Service Instance and Location for the Carrier Grade NAT, page 9
- Configuring the Service Virtual Interfaces, page 10

Configuring the Service Role

Perform this task to configure the service role on the specified location to start the CGN service.

Note Removal of service role is strictly not recommended while the card is active. This puts the card into FAILED state, which is service impacting.

SUMMARY STEPS

1. **configure**
2. **hw-module service cgn location** *node-id*
3. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**hw-module service cgn location** *node-id* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module service cgn location 0/1/CPU0`	Configures a CGN service role on location 0/1/CPU0.

Step 3	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Service Instance and Location for the Carrier Grade NAT

Perform this task to configure the service instance and location for the CGN application.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-location preferred-active** *node-id* [**preferred-standby** *node-id*]
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

	Command or Action	Purpose
Step 3	**service-location preferred-active** *node-id* [**preferred-standby** *node-id*] **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-location preferred-active 0/1/CPU0 preferred-standby 0/4/CPU0`	Configures the active and standby locations for the CGN application.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Service Virtual Interfaces

Configuring the Infrastructure Service Virtual Interface

Perform this task to configure the infrastructure service virtual interface (SVI) to forward the control traffic. The subnet mask length must be at least 30 (denoted as /30). CGSE uses SVI and it is therefore recommended that access control list (ACL) be configured to protect it from any form of denial of service attacks. For a sample ACL configuration, see Configuring ACL for a Infrastructure Service Virtual Interface, page 39.

Note Do not remove or modify service infra interface configuration when the card is in Active state. The configuration is service affecting and the line card must be reloaded for the changes to take effect.

SUMMARY STEPS

1. **configure**
2. **interface ServiceInfra** *value*

3. **service-location** *node-id*
4. **ipv4 address** *address/mask*
5. **end**
   or
   commit
6. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**interface ServiceInfra** *value*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# interface ServiceInfra 1`
	`RP/0/RP0/CPU0:router(config-if)#`
		Configures the infrastructure service virtual interface (SVI) as 1 and enters CGN configuration mode.

Step 3	**service-location** *node-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# service-location 0/1/CPU0`	Configures the location of the CGN service for the infrastructure SVI.
Step 4	**ipv4 address** *address/mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.1.1.1/30`	Sets the primary IPv4 address for an interface.

	Command or Action	Purpose

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 6	**reload**
	
	Example:
	`RP/0/RP0/CPU0:Router#hw-mod location 0/3/cpu0 reload`
		Once the configuration is complete, the card must be reloaded for changes to take effect.
		`WARNING: This will take the requested node out of service.`
		`Do you wish to continue?[confirm(y/n)] y`

Configuring the Application Service Virtual Interface

Perform this task to configure the application service virtual interface (SVI) to forward data traffic.

SUMMARY STEPS

1. **configure**
2. **interface ServiceApp** *value*
3. **service cgn** *instance-name* **service-type nat44**
4. **vrf** *vrf-name*
5. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**interface ServiceApp** *value*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# interface ServiceApp 1`
	`RP/0/RP0/CPU0:router(config-if)#`
		Configures the application SVI as 1 and enters interface configuration mode.

Step 3	**service cgn** *instance-name* **service-type** *nat44* **Example:** `RP/0/RP0/CPU0:router(config-if)# service cgn cgn1`	Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.
Step 4	**vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# vrf insidevrf1`	Configures the VPN routing and forwarding (VRF) for the Service Application interface

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Service Type Keyword Definition

Perform this task to configure the service type key definition.

SUMMARY STEPS

1. **configure**
2. **service cgn nat44** *instance-name*
3. **service-type nat44 nat1**
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn nat44** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN NAT44 application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring an Inside and Outside Address Pool Map

Perform this task to configure an inside and outside address pool map with the following scenarios:

- The designated address pool is used for CNAT.
- One inside VRF is mapped to only one outside VRF.
- Multiple non-overlapping address pools can be used in a specified outside VRF mapped to different inside VRF.
- Max Outside public pool per CGSE/CGN instance is 64 K or 65536 addresses. That is, if a /16 address pool is mapped, then we cannot map any other pool to that particular CGSE.
- Multiple inside vrf cannot be mapped to same outside address pool.
- While Mapping Outside Pool Minimum value for prefix is 16 and maximum value is 26.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **map** [**outside-vrf** *outside-vrf-name*] **address-pool** *address/prefix*
6. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

	Command or Action	Purpose

Step 4	**inside-vrf** *vrf-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named insidevrf1 and enters CGN inside VRF configuration mode.

Step 5	**map** [**outside-vrf** *outside-vrf-name*] **address-pool** *address/prefix*
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# map outside-vrf outside vrf1 address-pool 10.10.0.0/16`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# map address-pool 100.1.0.0/16`
		Configures an inside VRF to an outside VRF and address pool mapping.

Step 6	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-afi)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-afi)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Policy Functions for the Carrier Grade NAT

Configuring the Port Limit Per Subscriber

Perform this task to configure the port limit per subscriber for the system that includes TCP, UDP, and ICMP.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **portlimit** *value*
5. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

	Command or Action	Purpose
Step 4	**portlimit** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat44)# portlimit 10`	Limits the number of entries per address for each subscriber of the system

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Timeout Value for the Protocol

Configuring the Timeout Value for the ICMP Protocol

Perform this task to configure the timeout value for the ICMP type for the CGN instance.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **protocol icmp**
5. **timeout** *seconds*
6. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**protocol icmp**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# protocol icmp`
	`RP/0/RP0/CPU0:router(config-cgn-proto)#`
		Configures the ICMP protocol session. The example shows how to configure the ICMP protocol for the CGN instance named cgn1.

Step 5	**timeout** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-cgn-proto)# timeout 908`	Configures the timeout value as 908 for the ICMP session for the CGN instance named cgn1.

Step 6	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-proto)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-proto)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Timeout Value for the TCP Session

Perform this task to configure the timeout value for either the active or initial sessions for TCP.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **protocol tcp**
5. **session** {**active** | **initial**} **timeout** *seconds*
6. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**protocol tcp**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# protocol tcp`
	`RP/0/RP0/CPU0:router(config-cgn-proto)#`
		Configures the TCP protocol session. The example shows how to configure the TCP protocol for the CGN instance named cgn1.

	Command or Action	Purpose
Step 5	**session** {**active** | **initial**} **timeout** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-cgn-proto)# session initial timeout 90`	Configures the timeout value as 90 for the TCP session. The example shows how to configure the initial session timeout.

Step 6	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-proto)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-proto)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Timeout Value for the UDP Session

Perform this task to configure the timeout value for either the active or initial sessions for UDP.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **protocol udp**
5. **session** {**active** | **initial**} **timeout** *seconds*
6. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**protocol udp**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# protocol udp`
	`RP/0/RP0/CPU0:router(config-cgn-proto)#`
		Configures the UDP protocol sessions. The example shows how to configure the TCP protocol for the CGN instance named cgn1.

Step 5	**session** {**active** | **initial**} **timeout** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-cgn-proto)# session active timeout 90`	Configures the timeout value as 90 for the UDP session. The example shows how to configure the active session timeout.

Step 6	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-proto)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-proto)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Application Level Gateway

Perform this task to configure the application level gateway (ALG) for the active FTP connection for the specified CGN instance. The active FTP connection can be initiated from inside to an outside server with the FTP ALG enabled. Only the ActiveFTP is supported.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **alg ActiveFTP**
5. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

	Command or Action	Purpose

Step 4	**alg ActiveFTP**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# alg ActiveFTP`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the active ALG on the CGN instance named cgn1.

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the TCP Adjustment Value for the Maximum Segment Size

Perform this task to configure the adjustment value for the maximum segment size (MSS) for the VRF. You can configure the TCP MSS adjustment value on each VRF.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **protocol tcp**
6. **mss** *size*
7. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)#`
	`service-location preferred-active 0/1/CPU0`
	`preferred-standby 0/4/CPU0`
		Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)#`
	`inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**protocol tcp**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
	`protocol tcp`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-proto)#`
		Configures the TCP protocol session and enters CGN inside VRF AFI protocol configuration mode.

	Command or Action	Purpose
Step 6	**mss** *size* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-afi-proto )# mss 1100`	Configures the adjustment MSS value as 1100 for the inside VRF.

Step 7	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-proto)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-proto)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Refresh Direction for the Network Address Translation

Perform this task to configure the NAT mapping refresh direction as outbound for TCP and UDP traffic.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **refresh-direction Outbound**
5. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**refresh-direction Outbound**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# protocol tcp`
	`RP/0/RP0/CPU0:router(config-cgn-proto)#refresh-direction Outbound`
		Configures the NAT mapping refresh direction as outbound for the CGN instance named cgn1.

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Carrier Grade NAT for Static Port Forwarding

Perform this task to configure CGN for static port forwarding for reserved or nonreserved port numbers.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **protocol tcp**
6. **static-forward inside**
7. **address** *address* **port** *number*
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**protocol tcp**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# protocol tcp`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-proto)#`
		Configures the TCP protocol session and enters CGN inside VRF AFI protocol configuration mode.

	Command or Action	Purpose

Step 6	**static-forward inside**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-proto)# static-forward inside`
	`RP/0/RP0/CPU0:router(config-cgn-ivrf-sport-inside)#`
		Configures the CGN static port forwarding entries on reserved or nonreserved ports and enters CGN inside static port inside configuration mode.

Step 7	**address** *address* **port** *number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-ivrf-sport-inside)# address 1.2.3.4 port 90`	Configures the CGN static port forwarding entries for the inside VRF.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-ivrf-sport-inside)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-ivrf-sport-inside)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Export and Logging for the Network Address Translation Table Entries

Configuring the Server Address and Port for Netflow Logging

Perform this task to configure the server address and port to log network address translation (NAT) table entries for Netflow logging.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflowv9**
6. **server**
7. **address** *address* **port** *number*
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**external-logging netflowv9**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflowv9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGN instance named cgn1 and enters CGN inside VRF address family external logging configuration mode.

	Command or Action	Purpose

Step 6	**server**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflowv9-based external-logging facility and enters CGN inside VRF address family external logging server configuration mode.

Step 7	**address** *address* **port** *number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# address 2.3.4.5 port 45`	Configures the IPv4 address and port number 45 to log Netflow entries for the NAT table.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Path Maximum Transmission Unit for Netflow Logging

Perform this task to configure the path maximum transmission unit (MTU) for the netflowv9-based external-logging facility for the inside VRF.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflowv9**
6. **server**
7. **path-mtu** *value*

8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**external-logging netflowv9**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflowv9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGN instance named cgn1 and enters CGN inside VRF address family external logging configuration mode.

Step 6	**server**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflowv9-based external-logging facility and enters CGN inside VRF address family external logging server configuration mode.

	Command or Action	Purpose
Step 7	**path-mtu** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# path-mtu 2900`	Configures the path MTU with the value of 2900 for the netflowv9-based external-logging facility.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Refresh Rate for Netflow Logging

Perform this task to configure the refresh rate at which the Netflow-v9 logging templates are refreshed or resent to the Netflow-v9 logging server.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflowv9**
6. **server**
7. **refresh-rate** *value*
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**external-logging netflowv9**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflowv9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGN instance named cgn1 and enters CGN inside VRF address family external logging configuration mode.

Step 6	**server**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflow-v9 based external-logging facility and enters CGN inside VRF address family external logging server configuration mode.

	Command or Action	Purpose
Step 7	**refresh-rate** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# refresh-rate 50`	Configures the refresh rate value of 50 to log Netflow-based external logging information for an inside VRF.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Timeout for Netflow Logging

Perform this task to configure the frequency in minutes at which the Netflow-V9 logging templates are to be sent to the Netflow-v9 logging server.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflowv9**
6. **server**
7. **timeout** *value*
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**external-logging netflowv9**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflowv9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGN instance named cgn1 and enters CGN inside VRF address family external logging configuration mode.

Step 6	**server**
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflowv9-based external-logging facility and enters CGN inside VRF address family external logging server configuration mode.

	Command or Action	Purpose
Step 7	**timeout** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# timeout 50`	Configures the timeout value of 50 for Netflow logging of NAT table entries for an inside VRF.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for Implementing the Carrier Grade NAT

This section provides the following configuration examples for CGN:

Configuring a Different Inside VRF Map to a Different Outside VRF: Example

The following example shows how to configure a different inside VRF map to a different outside VRF and different outside address pools:

```
service cgn cgn1
inside-vrf insidevrf1
map outside-vrf outsidevrf1 address-pool 100.1.1.0/24
!
!
inside-vrf insidevrf2
map outside-vrf outsidevrf2 address-pool 100.1.2.0/24
!
service-location preferred-active 0/2/cpu0 preferred-standby 0/3/cpu0
!
interface ServiceApp 1
```

```
vrf insidevrf1
ipv4 address 210.1.1.1 255.255.255.0
service cgn cgn1
!
router static
vrf insidevrf1
0.0.0.0/0 serviceapp 1
!
!
interface ServiceApp 2
vrf insidevrf2
ipv4 address 211.1.1.1 255.255.255.0
service cgn cgn1
service-type nat44 nat1
!
router static
vrf insidevrf2
0.0.0.0/0 serviceapp 2
!
!
interface ServiceApp 3
vrf outsidevrf1
ipv4 address 1.1.1.1 255.255.255.0
service cgn cgn1
service-type nat44 nat1
!
router static
vrf outsidevrf1
100.1.1.0/24 serviceapp 3
!
!
interface ServiceApp 4
vrf outsidevrf2
ipv4 address 2.2.2.1 255.255.255.0
service cgn cgn1
service-type nat44 nat1
!
router static
vrf outsidevrf2
100.1.2.0/24 serviceapp 4
```

Configuring a Different Inside VRF Map to a Same Outside VRF: Example

The following example shows how to configure a different inside VRF map to the same outside VRF but with different outside address pools:

```
service cgn cgn1
inside-vrf insidevrf1
map outside-vrf outsidevrf1 address-pool 100.1.1.0/24
!
inside-vrf insidevrf2
map outside-vrf outsidevrf1 address-pool 200.1.1.0/24
!
!
service-location preferred-active 0/2/cpu0 preferred-standby 0/3/cpu0
!
interface ServiceApp 1
vrf insidevrf1
ipv4 address 1.1.1.1 255.255.255.0
service cgn cgn1
!
```

```
router static
vrf insidevrf1
0.0.0.0/0 serviceapp 1
!
!
interface ServiceApp 2
vrf insidevrf2
ipv4 address 2.1.1.1 255.255.255.0
service cgn cgn1
!
router static
vrf insidevrf2
0.0.0.0/0 serviceapp 2
!
!
interface ServiceApp 3
vrf outsidevrf1
ipv4 address 100.1.1.1 255.255.255.0
service cgn cgn1
!
router static
vrf outsidevrf1
100.1.1.0/24 serviceapp 3
200.1.1.0/24 serviceapp 3
!
```

Configuring ACL for a Infrastructure Service Virtual Interface

In the following example output, the IP address 1.1.1.1 is used by the SVI on the MSC side and IP address 1.1.1.2 is used in the CGSE PLIM.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 access-list ServiceInfraFilter
RP/0/RP0/CPU0:router(config)# 100 permit ipv4 host 1.1.1.1 any
RP/0/RP0/CPU0:router(config)# 101 permit ipv4 host 1.1.1.2 any

RP/0/RP0/CPU0:router(config)# interface ServiceInfra1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.1.1.1 255.255.255.192 service-location
0/1/CPU0
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group ServiceInfraFilter egress
```

Use the **show controllers services boot-params** command to verify the IP addresses of SVI and the CGSE PLIM.

```
RP/0/RP0/CPU0:router# show controllers services boot-params location 0/1/CPU0

================================================
Boot Params
================================================
Phase of implmentation   : 1
Application              : CGN
MSC ipv4 addddress       : 1.1.1.1
Octeon ipv4 addddress    : 1.1.1.2
ipv4netmask              : 255.255.255.252
```

Additional References

For additional information related to Implementing the Carrier Grade NAT, see the following references:

Related Documents

Related Topic	Document Title
Cisco IOS XR Carrier Grade NAT commands	*Cisco IOS XR Carrier Grade NAT Command Reference for the Cisco CRS Router.*
Cisco CRS router getting started material	*Cisco IOS XR Getting Started Guide*
Information about user groups and task IDs	*Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*

Standards

Standards[1]	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

1. Not all supported standards are listed.

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs[1]	Title
RFC 4787	*Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*
RFC 5382	*NAT Behavioral Requirements for TCP*
RFC 5508	*NAT Behavioral Requirements for ICMP*

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

INDEX

CGC	Cisco IOS XR Carrier Grade NAT Configuration Guide
HC	Cisco IOS XR Interface and Hardware Component Configuration Guide
IC	Cisco IOS XR IP Addresses and Services Configuration Guide
MCC	Cisco IOS XR Multicast Configuration Guide
MNC	Cisco IOS XR System Monitoring Configuration Guide
MPC	Cisco IOS XR MPLS Configuration Guide
NFC	Cisco IOS XR NetFlow Configuration Guide
QC	Cisco IOS XR Modular Quality of Service Configuration Guide
RC	Cisco IOS XR Routing Configuration Guide
SC	Cisco IOS XR System Security Configuration Guide
SMC	Cisco IOS XR System Management Configuration Guide
VPC	Cisco IOS XR Virtual Private Network Configuration Guide

Numerics

A

C

D

E

I

N

P

T