Match-in-VRF Support for NAT

The Match-in-VRF Support for NAT feature supports Network Address Translation (NAT) of packets that communicate between two hosts within the same VPN routing and forwarding (VRF) instance. In intra-VPN NAT, both the local and global address spaces for end hosts are isolated to their respective VPNs, and as a result, the translated addresses for the hosts overlap each other. The Match-in-VRF Support for NAT feature helps separate the address space for translated addresses among VPNs.

Restrictions for Match-in-VRF Support for NAT

- The Match-in-VRF Support for NAT feature is not supported on interface overload configuration.
- The **match-in-vrf** keyword for intra-VPN NAT is not supported with CGN.

Information About Match-in-VRF Support for NAT

Match-in-VRF Support for NAT

In Cisco IOS XE Release 3.5S and later releases, the Match-in-VRF Support for NAT feature supports NAT of packets that communicate between two hosts within the same VPN.

The VRF-aware NAT enables communication between hosts in the private address space in different VPN routing and forwarding (VRF) instances and common servers in the Internet or the global domain. Because IP addresses of the inside hosts overlap with each other, the VRF-aware NAT facilitates communication between these hosts by converting overlapped inside IP addresses into globally unique addresses. The Match-in-VRF Support for NAT feature extends VRF-aware NAT by supporting intra-VPN NAT capability. In the intra-VPN NAT, both the local and global address spaces for end hosts are isolated to their respective VPNs, and as a result translated addresses for hosts overlap each other. To separate the address space for translated addresses among VPNs, configure the **match-in-vrf** keyword in the NAT mapping (**ip nat inside**

source command) configuration. Both static and dynamic NAT configurations support the **match-in-vrf** keyword.

Note All NAT commands that support VRF support the **match-in-vrf** keyword. Because NAT outside rules (**ip nat outside source** command) support the match-in-VRF functionality by default, the **match-in-vrf** keyword is not supported by NAT outside rules.

In VRF-aware NAT, the IP alias and Address Resolution Protocol (ARP) entries for inside global addresses are configured in the global domain. For intra-VPN NAT, the IP alias and ARP entries for inside global addresses are configured in the VRF through which the translation happens. In intra-VPN NAT, configuration of the **match-in-vrf** keyword implies that at least one NAT outside interface is configured in the same VRF. The ARP entry in that VRF replies to the ARP request from the outside host.

If inside addresses are configured, the match-in-VRF is determined through inside mappings during the address translation of VRF traffic. If you have configured only outside mapping of IP addresses for address translations, the match-in-VRF will work. When a translation entry is created with both inside and outside mappings, the **match-in-vrf** keyword is determined by the inside mapping.

The Match-in-VRF Support for NAT feature supports the configuration of multiple dynamic mappings with the same IP address pool.

The following table provides you information about VRF support for NAT:

NAT Inside Interface	NAT Outside Interface
Global	Global IPv4 (non-MPLS)
MPLS IP	VRF **Note** You must use the **match-in-vrf** keyword in the configuration to indicate that communication is occurring within the VRF.
VRF	VRF **Note** Both VRFs must be in the same inside interface for this configuration to work.
VRF	MPLS **Note** You must use the **match-in-vrf** keyword in the configuration to indicate that communication is occurring within the VRF.
VRF	Global IPv4 (non-MPLS)

How to Configure Match-in-VRF Support for NAT

Configuring Static NAT with Match-in-VRF

Perform the following task to configure a static NAT translation and to enable NAT inside and outside traffic in the same VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**match-in-vrf**]]
4. **interface** *type number*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip nat inside**
7. **ip vrf forwarding** *vrf-name*
8. **exit**
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **ip nat outside**
12. **ip vrf forwarding** *vrf-name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**match-in-vrf**]] **Example:** `Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1 vrf vrf1 match-in-vrf`	Establishes static translation between an inside local address and an inside global address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 11	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside. **Note** NAT outside rules support the match-in-VRF functionality by default.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface.
Step 13	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Dynamic NAT with Match-in-VRF

Perform the following task to configure a dynamic NAT translation with the same address pool and to enable NAT inside and outside traffic in the same VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source list** *access-list-number* **pool** *pool-name* [**vrf** *vrf-name* [**match-in-vrf**]]
4. **access-list** *access-list-number* **permit source** [*source-wildcard*]
5. **ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name* [**match-in-vrf**]
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **ip vrf forwarding** *vrf-name*
10. **exit**
11. **interface** *type number*
12. **ip address** *ip-address mask*
13. **ip nat outside**
14. **ip vrf forwarding** *vrf-name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source list** *access-list-number* **pool** *pool-name* [**vrf** *vrf-name* [**match-in-vrf**]] **Example:** `Router(config)# ip nat inside source list 1 pool shared-pool vrf vrf1 match-in-vrf`	Enables multiple dynamic mappings to be configured with the same address pool. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.
Step 4	**access-list** *access-list-number* **permit source** [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated.
Step 5	**ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name* [**match-in-vrf**] **Example:** `Router(config)# ip nat inside source list 1 pool shared-pool vrf vpn1`	Establishes dynamic source translation, specifying the access list defined in the previous step.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters interface configuration mode.
Step 12	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 13	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside. **Note** NAT outside rules support the match-in-VRF functionality by default.
Step 14	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface.
Step 15	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Configuration Examples for Match-in-VRF Support for NAT

Example: Configuring Static NAT with Match-in-VRF

The following example shows how to configure a static NAT translation between the local IP address 10.10.10.1 and the global IP address 172.16.131.1. The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.

```
Router# configure terminal
Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1 vrf vrf1 match-in-vrf
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# ip address 10.114.11.39 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat outside
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# end
```

Example: Configuring Dynamic NAT with Match-in-VRF

The following example shows how to configure dynamic NAT mappings with the same address pool. The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.

```
Router# configure terminal
Router(config)# ip nat inside source list 1 pool shared-pool vrf vrf1 match-in-vrf
Router(config)# access-list 1 permit 192.168.34.0 0.0.0.255
Router(config)# ip nat inside source list 1 pool shared-pool vrf vpn1
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat inside
Router(config-if)# ip vrf forwarding vpn1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat outside
Router(config-if)# ip vrf forwarding vpn1
Router(config-if)# end
```

Additional References for Static NAT Mapping with HSRP

Related Documents

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

Related Topic	Document Title
NAT commands: complete command syntax, command mode, command history, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
IP Access List Sequence Numbering	*IP Access List Sequence Numbering* document
NAT configuration tasks	"Configuring NAT for IP Address Conservation" module
NAT maintenance	"Monitoring and Maintaining NAT" module
Using NAT with MPLS VPNs	"Integrating NAT with MPLS VPNs" module

Standards and RFCs

Standard/RFC	Title
RFC 903	*Reverse Address Resolution Protocol*
RFC 826	*Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware*
RFC 1027	*Using ARP to implement transparent subnet gateways*

Technical Assistance

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

Feature Information for Match-in-VRF Support for NAT

Table 1: Feature Information for Match-in-VRF Support for NAT

Feature Name	Releases	Feature Information
Match-in-VRF Support for NAT	Cisco IOS XE Release 3.5S	The Match-in-VRF Support for NAT feature supports the NAT translation of packets that communicate between two hosts within the same VPN.