Configure VRF-Aware Software Infrastructure NAT on IOS XE

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

- Working of VASI

Configure

- Network Diagram
- Initial Configurations
- VASI Interface Configuration
- NAT Configuration
  - Scenario 1 - NAT on Vasiright
  - Scenario 2 - NAT on Vasileft

Verify

Troubleshoot

Related Information

Introduction

This document describes the configuration of VRF-Aware Software Infrastructure (VASI) Network Address Translation (NAT) on routers that run Cisco IOS® XE.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions. This document applies to all Cisco routers and switches that run Cisco IOS XE.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

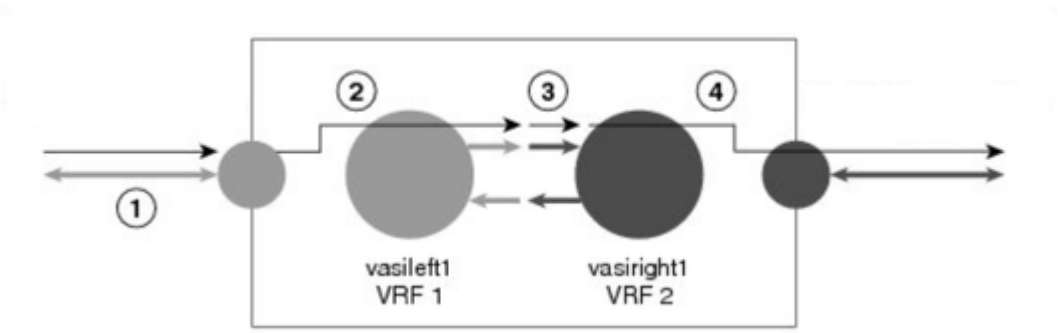
Background Information

Devices that run on Cisco IOS XE do not support classical inter-VRF NAT configurations as those found on Cisco IOS devices. Support for inter-VRF NAT on Cisco IOS XE is achieved via VASI implementation.

VASI provides the ability to configure services such as IPsec, firewall, and NAT to traffic that flows between virtual routing and forwarding (VRF) instances.

VASI is implemented by configuring VASI pairs, where each of the interfaces in the pair is associated with a different VRF instance. The VASI virtual interface is the next-hop interface for any packet that needs to be switched between these two VRF instances. The pairing is done automatically based on the two interface indexes such that the vasileft interface is automatically paired to the vasiright interface. Any packet that enters the vasileft interface is automatically forwarded to its paired vasiright interface.

Working of VASI



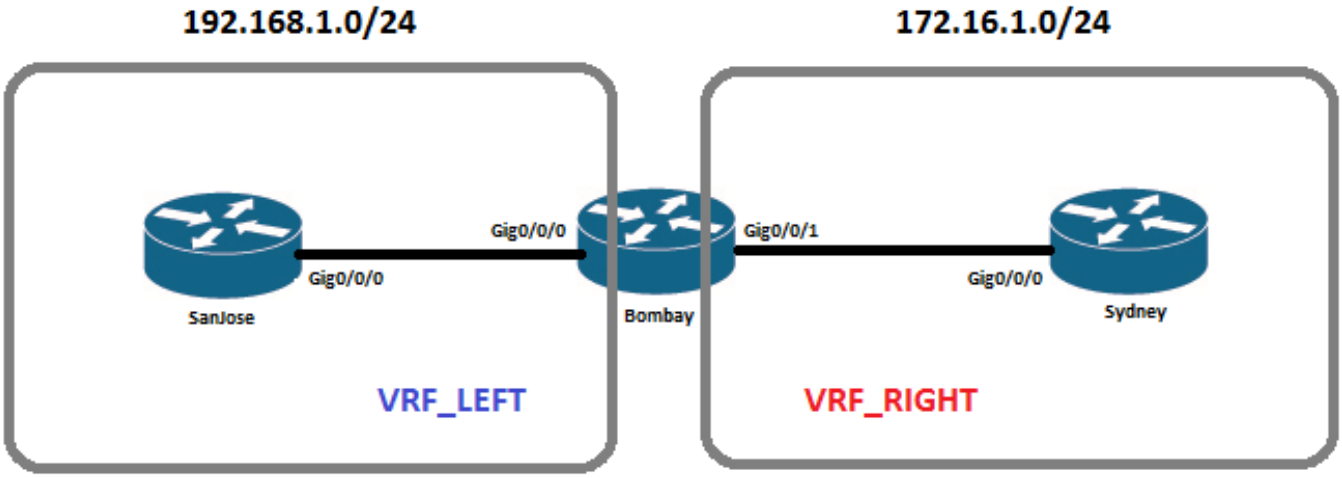
When an inter-VRF VASI is configured on the same device, the packet flow happens in this order:

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

Configure

These scenarios describe basic inter-VRF NAT configuration.

Network Diagram



Initial Configurations

SanJose:

```
interface GigabitEthernet0/0/0
 ip address 192.168.1.1 255.255.255.0

ip route 0.0.0.0 0.0.0.0 192.168.1.2
```

Mumbai:

```
vrf definition VRF_LEFT
 rd 1:1
 !
 address-family ipv4
 exit-address-family

vrf definition VRF_RIGHT
 rd 2:2
 !
 address-family ipv4
 exit-address-family

interface GigabitEthernet0/0/0
  vrf forwarding VRF_LEFT
  ip address 192.168.1.2 255.255.255.0

interface GigabitEthernet0/0/1
```

```
 vrf forwarding VRF_RIGHT
 ip address 172.16.1.2 255.255.255.0
```

Sydney:

```
interface GigabitEthernet0/0/0
 ip address 172.16.1.1 255.255.255.0
```

VASI Interface Configuration

Each VASI interface is paired to a different VRF instance.

```
interface vasileft1
 vrf forwarding VRF_LEFT
 ip address 10.1.1.1 255.255.255.252

interface vasiright1
 vrf forwarding VRF_RIGHT
 ip address 10.1.1.2 255.255.255.252
```

NAT Configuration

In this example, NAT is to be configured with these requirements:

1. Static NAT - Source IP of 192.168.1.1 is to be translated to 172.16.1.5.
2. Dynamic NAT - Source subnet of 192.168.1.0/24 is to be translated to 172.16.1.5.

Scenario 1 - NAT on Vasiright

In most cases, the WAN interface would be on the outgoing VRF, VRF_RIGHT in this topology. In such cases, NAT can be configured between the vasiright and the WAN interface; traffic coming in on the vasiright interface from vasileft is configured as NAT inside, while the WAN interface would be the NAT outside interface.

In this scenario, we use static routes to traffic between the VRFs. A static route for the destination 172.16.0.0 subnet is configured on VRF_LEFT pointing to the vasileft interface and another route for the source subnet 192.168.0.0 is configured on VRF_RIGHT pointing to the vasiright interface.



Note: Do not configure NAT to translate the source IP address to the WAN interface IP address;

---

the router treats return traffic to be destined to itself and does not forward traffic to the VASI interface.

Static NAT:

```
!--- Interface configuration

interface vasiright1
 vrf forwarding VRF_RIGHT
 ip address 10.1.1.2 255.255.255.252
 ip nat inside

interface GigabitEthernet0/0/1
 vrf forwarding VRF_RIGHT
 ip address 172.16.1.2 255.255.255.0
 ip nat outside


!--- Static route configuration

ip route vrf VRF_LEFT 172.16.0.0 255.255.0.0 vasileft1 10.1.1.2
ip route vrf VRF_RIGHT 192.168.0.0 255.255.0.0 vasiright1 10.1.1.1


!--- NAT configuration

ip nat inside source static 192.168.1.1 172.16.1.5 vrf VRF_RIGHT
```

Verification:

<#root>

```
Bombay#

sh ip nat translations vrf VRF_RIGHT

Pro         Inside global     Inside local       Outside local    Outside global
---         172.16.1.5        192.168.1.1        ---              ---
icmp        172.16.1.5:8      192.168.1.1:8      172.16.1.1:8     172.16.1.1:8
tcp         172.16.1.5:47491  192.168.1.1:47491  172.16.1.1:23    172.16.1.1:23
Total number of translations: 3
```

Dynamic NAT:

```
!--- Interface configuration

interface vasiright1
 vrf forwarding VRF_RIGHT
```

```
 ip address 10.1.1.2 255.255.255.252
 ip nat inside

interface GigabitEthernet0/0/1
 vrf forwarding VRF_RIGHT
 ip address 172.16.1.2 255.255.255.0
 ip nat outside


!--- Static route configuration

ip route vrf VRF_LEFT 172.16.0.0 255.255.0.0 vasileft1 10.1.1.2
ip route vrf VRF_RIGHT 192.168.0.0 255.255.0.0 vasiright1 10.1.1.1


!--- Access-list configuration
Extended IP access list 100
 10 permit tcp 192.168.1.0 0.0.0.255 host 172.16.1.1
 20 permit udp 192.168.1.0 0.0.0.255 host 172.16.1.1
 30 permit icmp 192.168.1.0 0.0.0.255 host 172.16.1.1


!--- NAT configuration

ip nat pool POOL 172.16.1.5 172.16.1.5 prefix-length 24
ip nat inside source list 100 pool POOL vrf VRF_RIGHT overload
```



Note: Configuring both the VASI interfaces of a pair as outside is not supported.

Verification:

<#root>

```
Bombay#

sh ip nat translations

Pro      Inside global      Inside local         Outside local      Outside global
icmp     172.16.1.5:1       192.168.1.1:15       172.16.1.1:15      172.16.1.1:1
tcp      172.16.1.5:1024    192.168.1.1:58166    172.16.1.1:23      172.16.1.1:23
Total number of translations: 2
```

Scenario 2 - NAT on Vasileft

NAT can also be configured solely on the vasileft side,that is VRF_LEFT and have traffic NATted before it is sent to VRF_RIGHT. The incoming interface on VRF_LEFT is considered as the NAT inside interface, and vasileft 1 is configured as the NAT outside interface.

In this scenario, we use static routes to traffic between the VRFs. A static route for the destination

172.16.0.0 subnet is configured on VRF_LEFT pointing to the vasileft interface and another route for the source NATted IP 172.16.1.5 is configured on VRF_RIGHT pointing to the vasiright interface.

Static NAT:

```
!--- Interface configuration

interface GigabitEthernet0/0/0
 vrf forwarding VRF_LEFT
 ip address 192.168.1.2 255.255.255.0
 ip nat inside

interface vasileft1
 vrf forwarding VRF_LEFT
 ip address 10.1.1.1 255.255.255.252
 ip nat outside


!--- Static route configuration

ip route vrf VRF_LEFT 172.16.0.0 255.255.0.0 vasileft1 10.1.1.2
ip route vrf VRF_RIGHT 172.16.1.5 255.255.255.255 vasiright1 10.1.1.1


!--- NAT configuration
ip nat inside source static 192.168.1.1 172.16.1.5 vrf VRF_LEFT
```

Verification:

<#root>

```
Bombay#

sh ip nat translations vrf VRF_LEFT

Pro       Inside global        Inside local         Outside local      Outside global
---       172.16.1.5           192.168.1.1          ---                ---
icmp      172.16.1.5:5         192.168.1.1:5        172.16.1.1:5       172.16.1.1:5
tcp       172.16.1.5:35414     192.168.1.1:35414    172.16.1.1:23      172.16.1.1:23
Total number of translations: 3
```

Dynamic NAT:

```
!--- Interface configuration

interface GigabitEthernet0/0/0
 vrf forwarding VRF_LEFT
 ip address 192.168.1.2 255.255.255.0
 ip nat inside

interface vasileft1
```

```
 vrf forwarding VRF_LEFT
 ip address 10.1.1.1 255.255.255.252
 ip nat outside


!--- Static route configuration

ip route vrf VRF_LEFT 172.16.0.0 255.255.0.0 vasileft1 10.1.1.2
ip route vrf VRF_RIGHT 172.16.1.5 255.255.255.255 vasiright1 10.1.1.1


!--- Access-list configuration

Extended IP access list 100
 10 permit tcp 192.168.1.0 0.0.0.255 host 172.16.1.1
 20 permit udp 192.168.1.0 0.0.0.255 host 172.16.1.1
 30 permit icmp 192.168.1.0 0.0.0.255 host 172.16.1.1


!--- NAT configuration

ip nat pool POOL 172.16.1.5 172.16.1.5 prefix-length 24
ip nat inside source list 100 pool POOL vrf VRF_LEFT overload
```

Verification:

<#root>

```
Bombay#

sh ip nat translations vrf VRF_LEFT


Pro      Inside global      Inside local        Outside local    Outside global
icmp     172.16.1.5:1       192.168.1.1:4       172.16.1.1:4     172.16.1.1:1
tcp      172.16.1.5:1024    192.168.1.1:27593   172.16.1.1:23    172.16.1.1:23
Total number of translations: 2
```

Verify

Use this section in order to confirm that your configuration works properly.

1. Check if dynamic/static routes are configured to route traffic between the two VRF instances.
2. Check if NAT has been configured for the correct VRF.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.

Related Information

- Configuring the VRF-Aware Software Infrastructure
- Technical Support & Downloads - Cisco Systems