

# Configure a VRF Aware Site-to-Site Tunnel with IKEv2 on FTD

## Contents

---

### [Introduction](#)

### [Prerequisites](#)

[Requirements](#)

[Components Used](#)

### [Virtual Routing and Forwarding](#)

[Limitations](#)

[Limitation 1](#)

[Limitation 2](#)

### [Network Diagram](#)

### [Configure](#)

### [Route Leak](#)

[Route Leaking from VRF to Global](#)

[Route Leaking from Global to VRF](#)

### [Verify](#)

### [Troubleshoot](#)

---

## Introduction

This document describes how to configure Virtual Routing and Forwarding (VRF)-aware IKEv2 site-to-site VPN tunnel on Firepower Threat Defense (FTD) managed by a Firepower Management Centre (FMC).

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Basic understanding of VPN
- Experience with FMC
- Knowledge of VRF implementation

### Components Used

The information in this document is based on these software versions:

- Cisco FMC version 7.x
- Cisco FTD version 7.x



**Note:** 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.

---

## Virtual Routing and Forwarding

In virtual routing, you can create multiple virtual routers to maintain separate routing tables for groups of interfaces to achieve network separation. This increases functionality by segmenting network paths without the use of multiple devices.

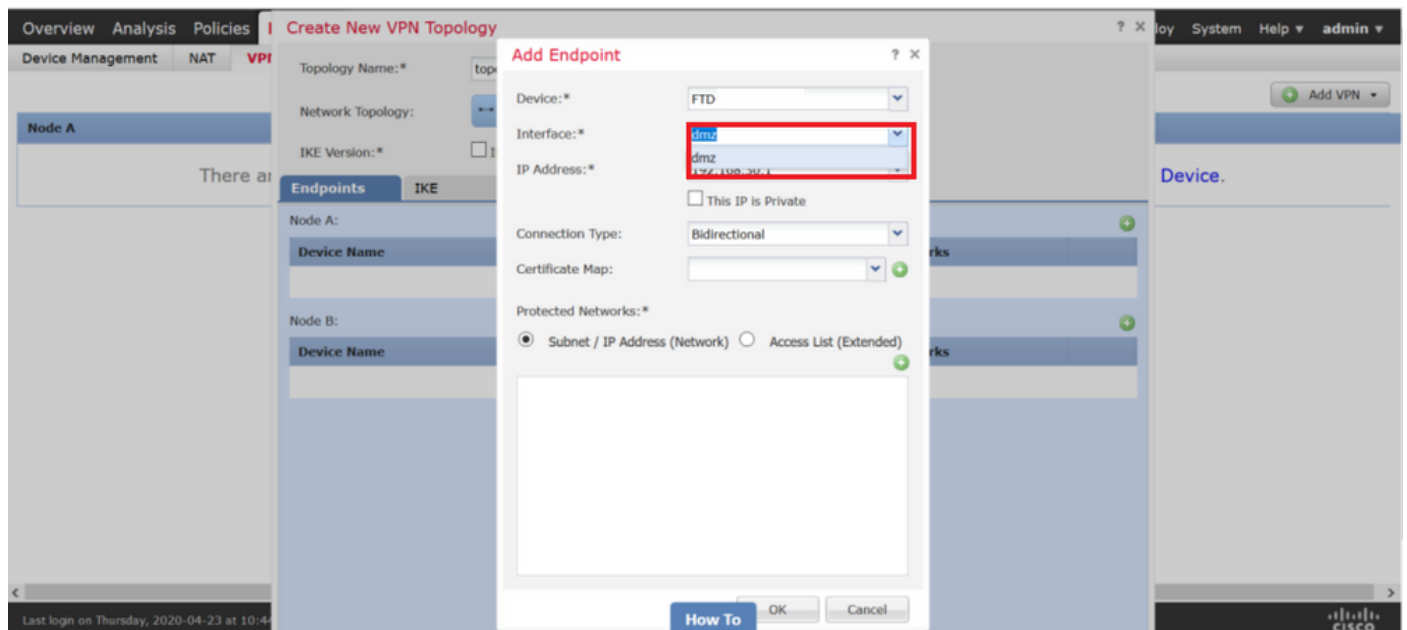
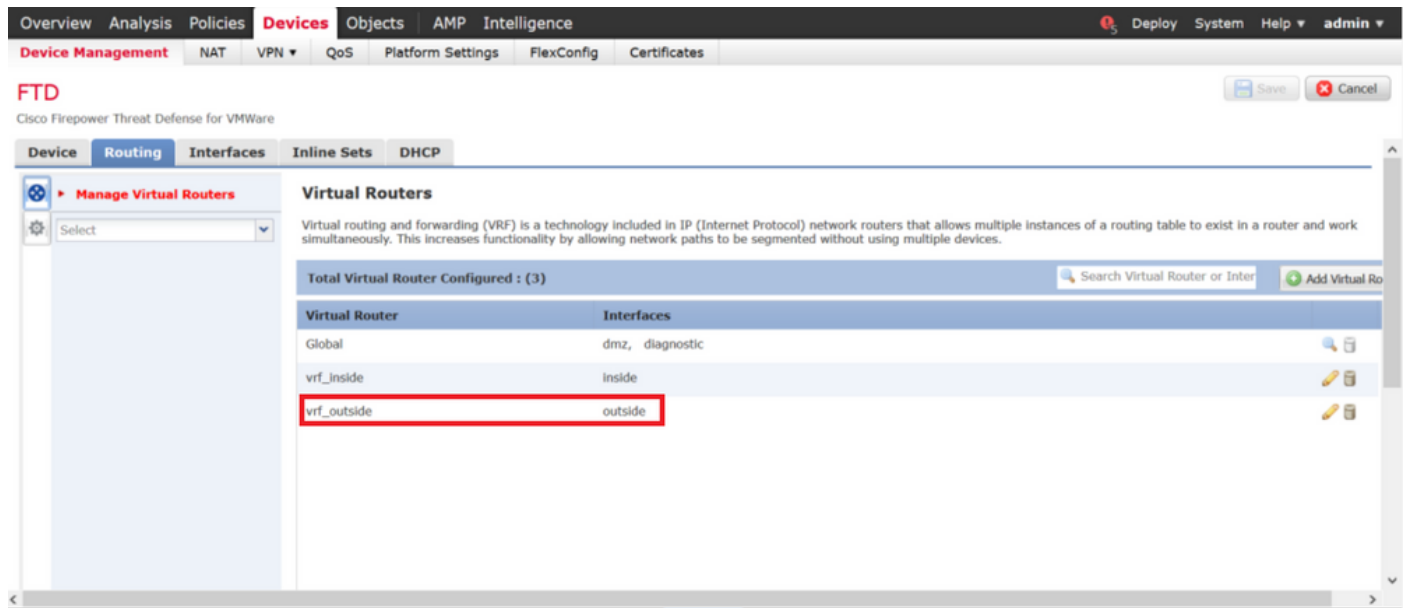
Because the routing instances are independent, IP addresses that overlap can be used without any conflict with each other. Each VRF has its own routing protocol sessions and IPv4, and IPv6 routing tables.

### Limitations

- The interface(s) which are in any VRF instance cannot be used as a tunnel endpoint/VPN interface.
- An interface used to terminate the VPN tunnel can only be in Global VRF.

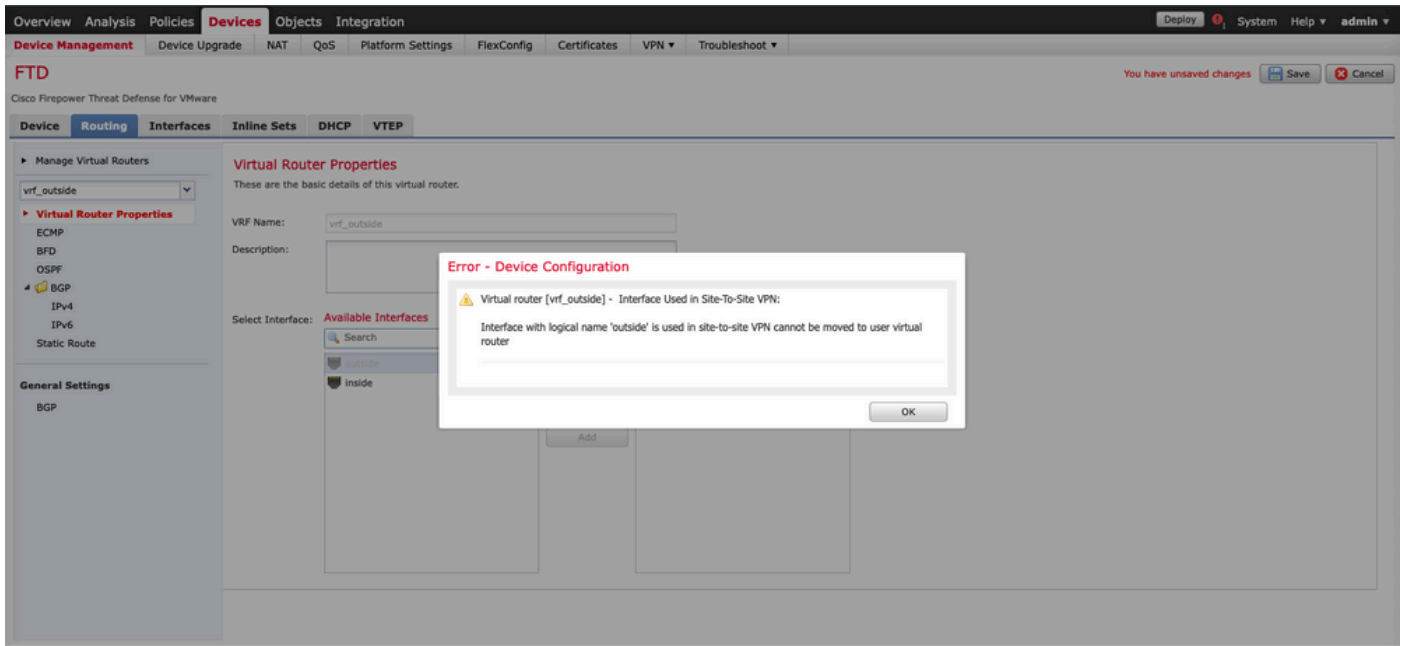
#### Limitation 1

If the **outside** interface is added to the virtual router **vrf\_outside**, then this interface is not shown in the dropdown for endpoint interface selection when a site-to-site VPN topology is created.

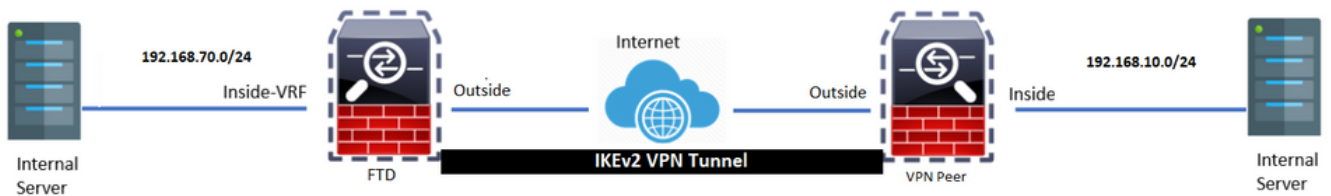


## Limitation 2

If a site-to-site VPN topology on the **outside** interface exists, then it is not possible to add the interface to a VRF instance. FMC gives an error that states that the **outside** (WAN) interface that acts as a VPN tunnel terminates the endpoint to be a part of Global VRF and not a custom VRF.



## Network Diagram



## Configure

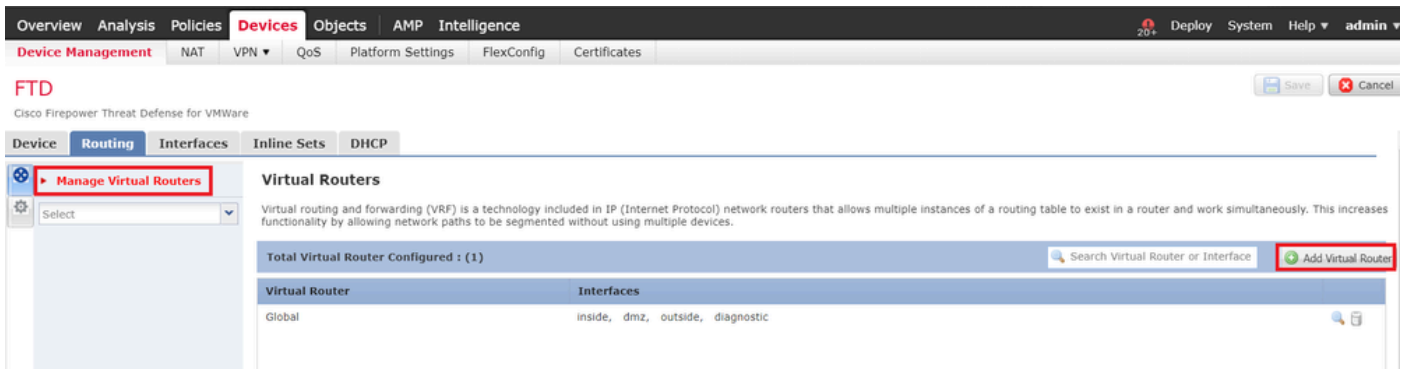
Configure an IKEv2 site-to-site VPN tunnel between FTD 7.x and any other device (ASA/FTD/Router or a third-party vendor).



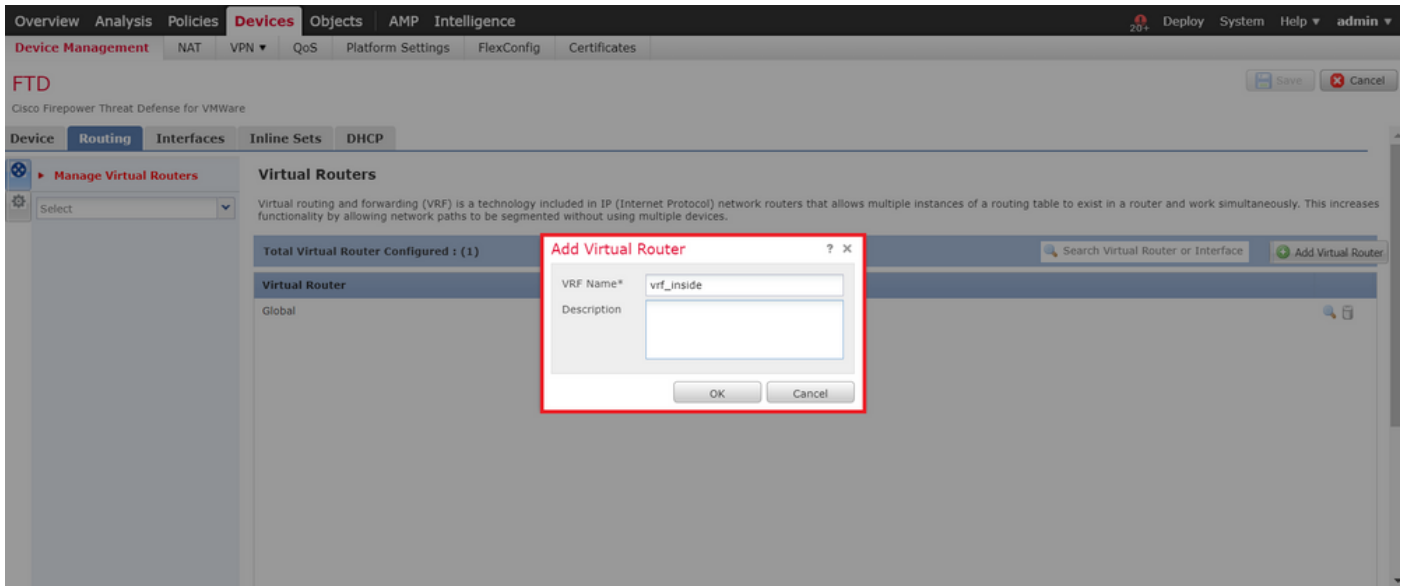
**Note:** This document assumes that site-to-site VPN tunnel is already configured. Please refer to [How to configure site-to-site VPN on FTD](#) managed by FMC for more details.

Navigate to **Devices > Device Management**. Click on **Edit** and then select **Routing**.

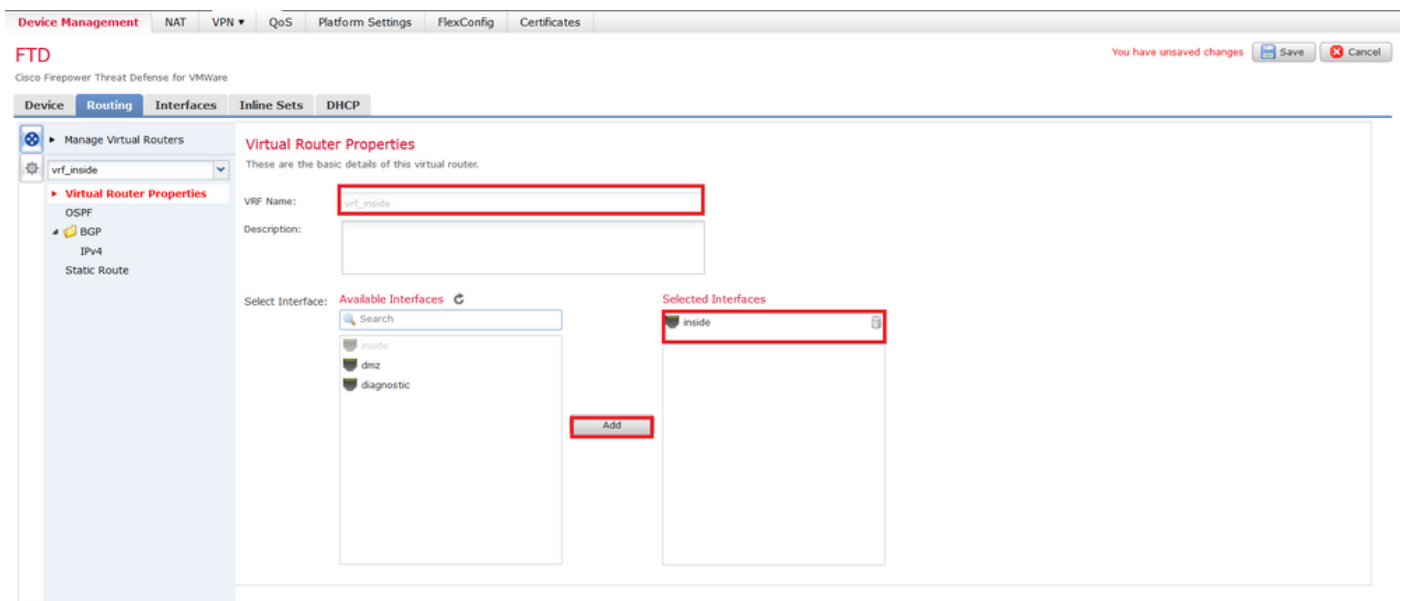
Step 1. Click on **Manage Virtual Routers** as shown in the image.



Step 2. Click on **Add Virtual Router** and add the required VRF instance to it. For this deployment, `vrf_inside` is used.



Step 3. Once the VRF instance is created, an option to add the required interface(s) is shown. For this deployment, inside interface is added to vrf\_inside as shown in the image.



Step 4. For this deployment, these are the traffic selectors for our site-to-site VPN tunnel.

Source: 192.168.70.0/24 [This network is on inside interface which is in "vrf\_inside"]

192.168.80.0/24 [This network is on dmz interface which is not in any vrf instance]

Destination : 192.168.10.0/24

## Route Leak

VRF allows a router to maintain separate routing tables for different virtual networks. When exceptions are needed, VRF route leaking allows some traffic to be routed between the VRFs. Route leaking between Global Routing Table (GRT) and Virtual Routing and Forwarding (VRF) table is done with the use of static

routes. Either method provides the next-hop IP address (for the multi-access segment) or points the route out of an interface (point-to-point interface).

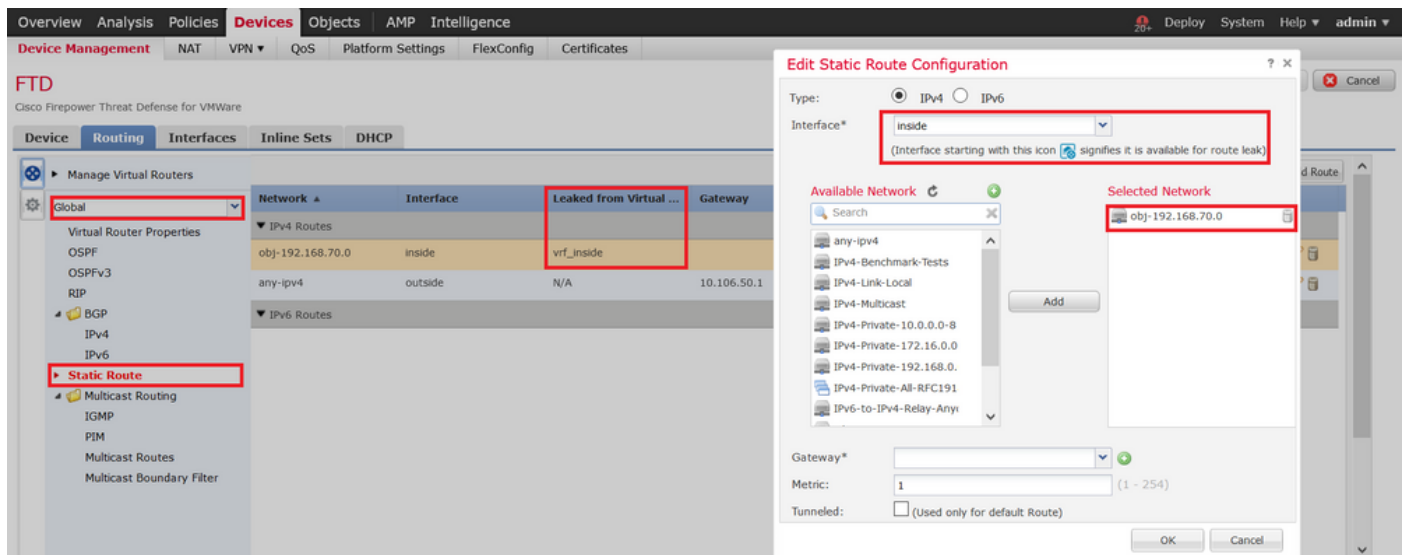
## Route Leaking from VRF to Global

1. Select Devices > Device Management, and click on Edit for FTD.
2. Click Routing. By default, the Global routing properties page appears.
3. Click Static Route.
4. Click Add Route, configure:

- Interface — Select the inside interface.
- Network — Select the vrf\_inside virtual router network object (192.168.70.0/24).
- Gateway — Leave it blank. When leaking a route into another virtual router, do not select the gateway.

The route leak allows endpoints protected by the external (remote) end of the site-to-site VPN to access the 192.168.70.0/24 network in the vrf\_inside virtual router.

5. Click OK as shown in the image.



On CLI, the route is shown as :

```
route inside 192.168.70.0 255.255.255.0 1
```

Note that the network 192.168.70.0/24 is directly connected to the inside interface but this network is not visible in GRT because the network is in the VRF instance. In order to make this route available in GRT, the route has been leaked from vrf\_inside to Global.

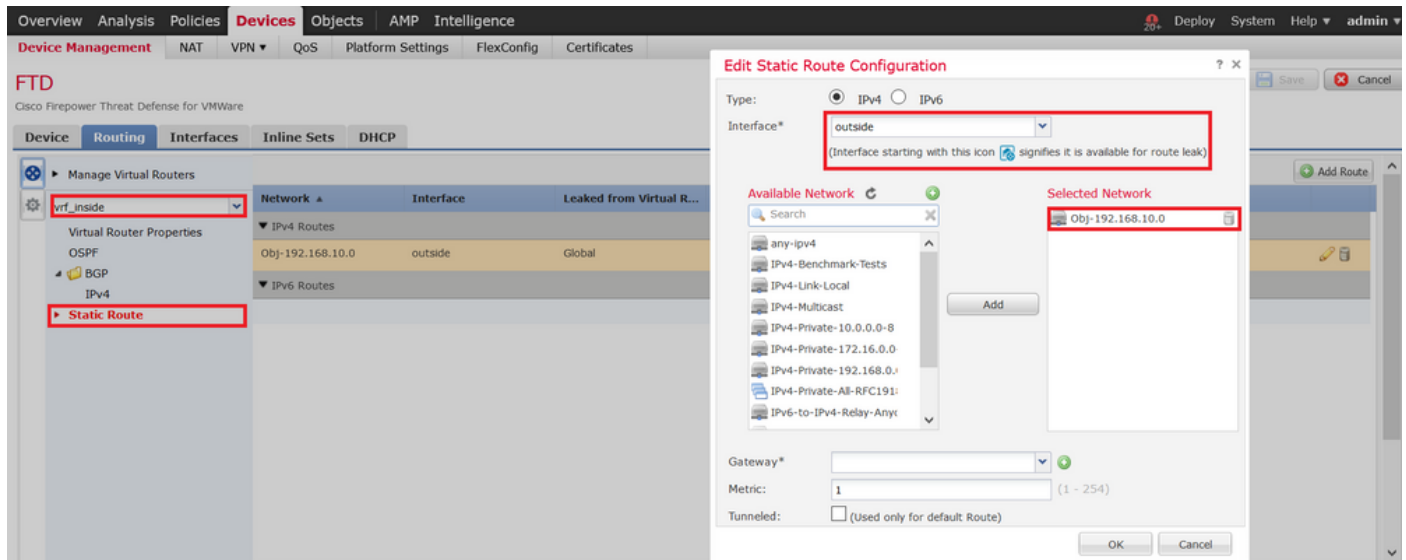
## Route Leaking from Global to VRF

1. Choose Devices > Device Management, and click on Edit.
2. Click Routing and from the drop-down, select vrf\_inside.
3. Click Static Route .
4. Click Add Route , configure:

- Interface — Select the outside interface of the global router
- Network — Select the global virtual router network object (192.168.10.0/24)
- Gateway — Leave it blank. When leaking a route into another virtual router, do not select the gateway

This static route allows endpoints on the 192.168.70.0/24 network to initiate connections to 192.168.10.0/24 that traverse through the site-to-site VPN tunnel.

5. Click ok as shown in the image.



On CLI, the route is shown as:

```
route vrf vrf_inside outside 192.168.10.0 255.255.255.0 1
```

## Verify

Use this section in order to confirm that your configuration works properly. All the outputs are collected from FTD shown in the network diagram.

```
FTD# show vrf
Name          VRF ID      Description          Interfaces
vrf_inside    1           vrf_inside          inside
```

```
FTD# show run route
route outside 10.0.0.0 255.0.0.0 10.106.50.1 1
route inside 192.168.70.0 255.255.255.0 1
```

```
FTD# show run route vrf vrf_inside
route vrf vrf_inside outside 192.168.10.0 255.255.255.0 1
```



FTD# show route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2, V - VPN  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, \* - candidate default, U - per-user static route  
o - ODR, P - periodic downloaded static route, + - replicated route  
Gateway of last resort is not set

```
S      10.0.0.0 255.0.0.0 [1/0] via 10.106.50.1, outside
C      10.106.50.0 255.255.255.0 is directly connected, outside
L      10.106.50.212 255.255.255.255 is directly connected, outside
V      192.168.10.0 255.255.255.0 connected by VPN (advertised), outside
S      192.168.70.0 255.255.255.0 [1/0] is directly connected, inside
C      192.168.80.0 255.255.255.0 is directly connected, dmz
L      192.168.80.1 255.255.255.255 is directly connected, dmz
```

FTD# show crypto ikev2 sa

IKEv2 SAs:

Session-id:8, Status:UP-ACTIVE, IKE count:1, CHILD count:1

```
Tunnel-id Local Remote
444445753 10.106.50.212/500 10.197.224.175/500
    Encr: AES-CBC, keysize: 256, Hash: SHA256, DH Grp:19, Auth sign: PSK, Auth verify: PSK
    Life/Active Time: 86400/11 sec
Child sa: local selector 192.168.70.0/0 - 192.168.70.255/65535
          remote selector 192.168.10.0/0 - 192.168.10.255/65535
          ESP spi in/out: 0x5e950adb/0x47acd2dc
```

FTD# show crypto ipsec sa peer 10.197.224.175

peer address: 10.197.224.175

Crypto map tag: CSM\_outside\_map, seq num: 2, local addr: 10.106.50.212

```
access-list vrf-crypto-acl extended permit ip 192.168.70.0 255.255.255.0 192.168.10.0 255.255.255.0
local ident (addr/mask/prot/port): (192.168.70.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (192.168.10.0/255.255.255.0/0/0)
current_peer: 10.197.224.175
```

```
#pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
#pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 4, #pkts comp failed: 0, #pkts decomp failed: 0
#pre-frag successes: 0, #pre-frag failures: 0, #fragments created: 0
#PMTUs sent: 0, #PMTUs rcvd: 0, #decapsulated frgs needing reassembly: 0
#TFC rcvd: 0, #TFC sent: 0
#Valid ICMP Errors rcvd: 0, #Invalid ICMP Errors rcvd: 0
#send errors: 0, #recv errors: 0
```

```
local crypto endpt.: 10.106.50.212/500, remote crypto endpt.: 10.197.224.175/500
path mtu 1500, ipsec overhead 74(44), media mtu 1500
PMTU time remaining (sec): 0, DF policy: copy-df
ICMP error validation: disabled, TFC packets: disabled
```

```
current outbound spi: 47ACD2DC
current inbound spi : 5E950ADB
```

#### inbound esp sas:

```
spi: 0x5E950ADB (1586825947)
SA State: active
transform: esp-aes-256 esp-sha-hmac no compression
in use settings ={L2L, Tunnel, IKEv2, }
slot: 0, conn_id: 10, crypto-map: CSM_outside_map
sa timing: remaining key lifetime (kB/sec): (4193279/28774)
IV size: 16 bytes
replay detection support: Y
Anti replay bitmap:
0x00000000 0x0000001F
```

#### outbound esp sas:

```
spi: 0x47ACD2DC (1202508508)
SA State: active
transform: esp-aes-256 esp-sha-hmac no compression
in use settings ={L2L, Tunnel, IKEv2, }
slot: 0, conn_id: 10, crypto-map: CSM_outside_map
sa timing: remaining key lifetime (kB/sec): (4147199/28774)
IV size: 16 bytes
replay detection support: Y
Anti replay bitmap:
0x00000000 0x00000001
```

## Troubleshoot

This section provides information you can use in order to troubleshoot your configuration.

```
<#root>
```

```
FTD# show crypto ipsec sa peer 10.197.224.175
```

```
peer address: 10.197.224.175
```

```
Crypto map tag: CSM_outside_map, seq num: 2, local addr: 10.106.50.212
```

```
access-list vrf-crypto-acl extended permit ip 192.168.70.0 255.255.255.0 192.168.10.0 255.255.255.0
local ident (addr/mask/prot/port): (192.168.70.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (192.168.10.0/255.255.255.0/0/0)
current_peer: 10.197.224.175
```

```
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
```

```
>>>> Packets received from remote end gets decapsulated but there are not encaps for the responses
```

```
#pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0
#pre-frag successes: 0, #pre-frag failures: 0, #fragments created: 0
#PMTUs sent: 0, #PMTUs rcvd: 0, #decapsulated frgs needing reassembly: 0
#TFC rcvd: 0, #TFC sent: 0
#Valid ICMP Errors rcvd: 0, #Invalid ICMP Errors rcvd: 0
#send errors: 0, #recv errors: 0
```

```
local crypto endpt.: 10.106.50.212/500, remote crypto endpt.: 10.197.224.175/500
path mtu 1500, ipsec overhead 74(44), media mtu 1500
PMTU time remaining (sec): 0, DF policy: copy-df
ICMP error validation: disabled, TFC packets: disabled
current outbound spi: 490F4CD1
```

current inbound spi : DB5608EB

inbound esp sas:

spi: 0xDB5608EB (3679848683)  
SA State: active  
transform: esp-aes-256 esp-sha-hmac no compression  
in use settings ={L2L, Tunnel, IKEv2, }  
slot: 0, conn\_id: 11, crypto-map: CSM\_outside\_map  
sa timing: remaining key lifetime (kB/sec): (4008959/28761)  
IV size: 16 bytes  
replay detection support: Y  
Anti replay bitmap:  
0x00000000 0x0000001F

outbound esp sas:

spi: 0x490F4CD1 (1225739473)  
SA State: active  
transform: esp-aes-256 esp-sha-hmac no compression  
in use settings ={L2L, Tunnel, IKEv2, }  
slot: 0, conn\_id: 11, crypto-map: CSM\_outside\_map  
sa timing: remaining key lifetime (kB/sec): (4239360/28761)  
IV size: 16 bytes  
replay detection support: Y  
Anti replay bitmap:  
0x00000000 0x00000001

<#root>

capture capin type raw-data interface inside [Capturing - 0 bytes]

>>>> Captures applied on LAN(inside) interface shows decapsulated packets are not routed into LAN network

match ip host 192.168.10.2 host 192.168.70.2

FTD# show cap capin

0 packet captured

0 packet shown

<#root>

capture asp type asp-drop all [Capturing - 0 bytes]

>>>> ASP Captures shows decapsulated packets are being dropped on FTD

FTD# show capture asp | i 192.168.70.2

145:	15:28:47.670894	192.168.10.2	>	192.168.70.2	icmp: echo request
154:	15:28:49.666545	192.168.10.2	>	192.168.70.2	icmp: echo request
171:	15:28:51.672740	192.168.10.2	>	192.168.70.2	icmp: echo request
172:	15:28:53.664928	192.168.10.2	>	192.168.70.2	icmp: echo request

<#root>

FTD# packet-tracer input outside icmp 192.168.10.2 8 0 192.168.70.2 detailed

>>>> Packet tracer from outside shows "no route" for 192.168.70.0/24 network

Phase: 1

Type: ACCESS-LIST

Subtype:  
Result: ALLOW  
Config:  
Implicit Rule  
Additional Information:  
Forward Flow based lookup yields rule:  
in id=0x2ba3bce77330, priority=1, domain=permit, deny=false  
hits=171480, user\_data=0x0, cs\_id=0x0, l3\_type=0x8  
src mac=0000.0000.0000, mask=0000.0000.0000  
dst mac=0000.0000.0000, mask=0100.0000.0000  
input\_ifc=outside, output\_ifc=any

Result:  
input-interface: outside(vrfid:0)  
input-status: up  
input-line-status: up  
Action: drop  
Drop-reason:

(no-route) No route to host  
, Drop-location: frame 0x000055d9b7e8c7ce flow (NA)/NA

<#root>

FTD# show run route  
route outside 10.0.0.0 255.0.0.0 10.106.50.1 1

>>>> As the network 192.168.70.0/24 is in "vrf\_inside" instance, there is no route leaked from Global to

<#root>

FTD# show run route  
route outside 10.0.0.0 255.0.0.0 10.106.50.1 1  
route inside 192.168.70.0 255.255.255.0 1

>>>> After leaking the route from Global to vrf\_inside

<#root>

FTD# show cap capin

>>>> Now capture shows bi-directional traffic on LAN(inside) interface

10 packets captured

1: 15:44:32.972743	192.168.10.2 > 192.168.70.2 icmp: echo request
2: 15:44:32.974543	192.168.70.2 > 192.168.10.2 icmp: echo reply
3: 15:44:33.032209	192.168.10.2 > 192.168.70.2 icmp: echo request
4: 15:44:33.033353	192.168.70.2 > 192.168.10.2 icmp: echo reply
5: 15:44:33.089656	192.168.10.2 > 192.168.70.2 icmp: echo request
6: 15:44:33.092814	192.168.70.2 > 192.168.10.2 icmp: echo reply
7: 15:44:33.149024	192.168.10.2 > 192.168.70.2 icmp: echo request
8: 15:44:33.151878	192.168.70.2 > 192.168.10.2 icmp: echo reply
9: 15:44:33.158774	192.168.10.2 > 192.168.70.2 icmp: echo request

10: 15:44:33.161048 192.168.70.2 > 192.168.10.2 icmp: echo reply  
10 packets shown

<#root>

FTD# packet-tracer input outside icmp 192.168.10.2 8 0 192.168.70.2 detailed

>>>> Verified packet flow using Packet tracer

Phase: 1  
Type: INPUT-ROUTE-LOOKUP  
Subtype: Resolve Egress Interface  
Result: ALLOW  
Config:  
Additional Information:  
Found next-hop 0.0.0.0 using egress ifc

inside(vrfid:1)

-----Output Omitted-----

Phase: 8  
Type: VPN  
Subtype: ipsec-tunnel-flow  
Result: ALLOW  
Config:  
Additional Information:  
Forward Flow based lookup yields rule:  
in id=0x2ba3bdc75cc0, priority=70, domain=ipsec-tunnel-flow, deny=false  
hits=7, user\_data=0xea71cdc, cs\_id=0x2ba3bce93e70, reverse, flags=0x0, protocol=0  
src ip/id=192.168.10.0, mask=255.255.255.0, port=0, tag=any  
dst ip/id=192.168.70.0, mask=255.255.255.0, port=0, tag=any, dscp=0x0  
input\_ifc=outside(vrfid:0), output\_ifc=any

-----Output Omitted-----

Phase: 13  
Type: VPN  
Subtype: encrypt  
Result: ALLOW  
Config:  
Additional Information:  
Reverse Flow based lookup yields rule:  
out id=0x2ba3bd44ed40, priority=70, domain=encrypt, deny=false  
hits=7, user\_data=0xea6e344, cs\_id=0x2ba3bce93e70, reverse, flags=0x0, protocol=0  
src ip/id=192.168.70.0, mask=255.255.255.0, port=0, tag=any  
dst ip/id=192.168.10.0, mask=255.255.255.0, port=0, tag=any, dscp=0x0  
input\_ifc=any

(vrfid:65535)

, output\_ifc=outside

Result:  
input-interface: outside(vrfid:0)  
input-status: up  
input-line-status: up

output-interface: inside(vrfid:1)

output-status: up

output-line-status: up

Action: drop

Drop-reason: (ipsec-spoof) IPSEC Spoof detected, Drop-location: frame 0x000055d9b7e8b4d1 flow (NA)/NA