|  |
| --- |
|  |
|  |
| vrf lab  Made by: - Akshat Kansal |
| **CCNP lab 7 – Mr. Mason & Mr. Hansen**  **Periods 0,1,2** |

**Purpose**

This lab is designed to offer network professionals an opportunity to gain practical knowledge and hands-on experience in using VRF technology to partition a single physical device into multiple virtual devices. By showcasing the key features and benefits of VRF technology, this lab aims to enable network administrators to optimize network performance, enhance security, and improve resource utilization in their organizations.

**Background**

Virtual Routing and Forwarding (VRF) is a technique used to virtualize routing tables in a layer-3 network. This allows for multiple routing tables to exist in a single physical router and work simultaneously. Each VRF has its own layer-3 forwarding table, and any device in a specific VRF can be layer-3 directly routed to another device in the same VRF but cannot directly reach one in another VRF.

This is similar to how VLANs work, where each VLAN has its own layer-2 forwarding and flooding domain, and devices in the same VLAN can communicate with each other at layer-2, but devices in different VLANs require a layer-3 router to forward traffic between them.

In the same way that VLANs are configured on a switch with each access port assigned to a specific VLAN, VRFs are assigned to layer-3 interfaces on a router. However, since VRF is a layer-3 virtualization technique, we also need to define routes with their particular next-hops to VRFs.

The main benefit of VRF is that it creates separate pockets in a router, allowing businesses to transport confidential information and data without the threat of interception by a rival business using the same ISP. This is especially important if the rival business is running packet-detecting programs or network protocol analyzers like Wireshark to intercept sensitive data.

When we reference layer-2, it usually involves either a switch or a MAC address, which is also known as a physical address. On the other hand, layer-3 involves a router or an IP address. By using VRF, businesses can create isolated routing tables and forwarding domains, which enhances security and allows for greater flexibility in network design.

**Lab summary**

I physically connected three routers in a linear arrangement using wires and connected one computer to each end of the router line. To distinguish the two groups of devices, I created two separate virtual routing and forwarding (VRF) groups for each router. Specifically, I placed the loopback interfaces in the "Microsoft" group and the computers in the "Google" group. By doing this, the devices in each group were isolated from each other, meaning that devices in the "Microsoft" group could only communicate with other devices in the same group and not with devices in the "Google" group.

To enable routing between the routers and the devices, I configured one Open Shortest Path First (OSPF) area for the entire network. To ensure that each router had a unique identifier, I added two network statements on each router. I also configured two sub-interfaces on each router, so that each sub-interface could be assigned to a different VRF group.

In summary, I set up a network topology that consisted of three routers and two computers, connected in a linear configuration. Each router was virtually split into two VRF groups, with the loopback interfaces in the "Microsoft" group and the computers in the "Google" group. I also configured OSPF for the entire network and two sub-interfaces on each router for each VRF group. This setup allowed for isolated communication between the devices in each VRF group, while still enabling routing between the routers and the devices.

**Lab commands**

The first command, "ip vrf {vrf-name}" is used to create a VRF table and enter into VRF configuration mode. This allows for the creation of separate virtual routing tables within a router.

The second command, "ip vrf forwarding {vrf-name}" is used to assign an interface to a specific VRF group. This enables the forwarding of packets to the correct VRF table based on the VRF forwarding table.

The third command, "show ip route vrf {vrf-name}" is used to display the routing table for a specific VRF. This allows for the examination of the routes that exist within a particular VRF group.

The fourth command, "ping vrf {vrf-name} ip {ip-address}" is used to perform a ping test to an IP address within a specific VRF. This allows for the testing of connectivity within a specific VRF group.

The fifth command, "router ospf {as-number} vrf {vrf-name}" enables OSPF processing for a specific VRF table. This allows OSPF to work with multiple VRF tables, enabling more complex network topologies.

**Pings**

R1#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#00:1::1/64

% Bad IP address or host name% Unknown command or computer name, or unable to find computer address

R1#ping 10.0.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#0:1::1/64

% Bad IP address or host name% Unknown command or computer name, or unable to find computer address

R1#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#ping 10.0.0.2

Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#ping 192.168.3.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#ping 10.0.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#ping 10.0.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#g 10:2::2/64

^

% Invalid input detected at '^' marker.

R1#ping 192.168.4.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#ping 192.168.5.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.5.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

**router 1**

hostname R1

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip vrf Google

ip vrf Microsoft

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214421CF

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

vlan 10,20

interface Loopback0

ip vrf forwarding Microsoft

ip address 192.168.1.3 255.255.255.0

interface GigabitEthernet0/0/0

no ip address

negotiation auto

interface GigabitEthernet0/0/0.10

encapsulation dot1Q 1 native

ip vrf forwarding Google

ip address 192.168.2.1 255.255.255.0

interface GigabitEthernet0/0/0.20

encapsulation dot1Q 2

ip vrf forwarding Microsoft

ip address 192.168.2.2 255.255.255.0

interface GigabitEthernet0/0/1

ip vrf forwarding Google

ip address 192.168.1.2 255.255.255.0

negotiation auto

interface Serial0/1/0

no ip address

interface Serial0/1/1

no ip address

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

negotiation auto

interface Vlan1

no ip address

router ospf 2 vrf Google

network 192.168.1.0 0.0.0.255 area 0

network 192.168.2.0 0.0.0.255 area 0

router ospf 3 vrf Microsoft

network 192.168.1.0 0.0.0.255 area 0

network 192.168.2.0 0.0.0.255 area 0

router ospf 1

router-id 1.1.1.1

network 192.168.1.0 0.0.0.255 area 0

network 192.168.2.0 0.0.0.255 area 0

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

**router 2:**

hostname R2

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip vrf Google

ip vrf Microsoft

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO211216BL

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

no ip address

negotiation auto

interface GigabitEthernet0/0/0.10

encapsulation dot1Q 1 native

ip vrf forwarding Google

ip address 192.168.3.20 255.255.255.0

interface GigabitEthernet0/0/0.20

encapsulation dot1Q 2

ip vrf forwarding Microsoft

ip address 192.168.3.21 255.255.255.0

interface GigabitEthernet0/0/1

no ip address

negotiation auto

interface GigabitEthernet0/0/1.10

encapsulation dot1Q 1 native

ip vrf forwarding Google

ip address 192.168.2.10 255.255.255.0

interface GigabitEthernet0/0/1.20

encapsulation dot1Q 2

ip vrf forwarding Microsoft

ip address 192.168.2.11 255.255.255.0

interface Serial0/1/0

no ip address

interface Serial0/1/1

no ip address

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

negotiation auto

interface Vlan1

no ip address

router ospf 1 vrf Google

network 192.168.2.0 0.0.0.255 area 0

network 192.168.3.0 0.0.0.255 area 0

router ospf 3 vrf Microsoft

network 192.168.2.0 0.0.0.255 area 0

network 192.168.3.0 0.0.0.255 area 0

router ospf 2

router-id 2.2.2.2

network 192.168.2.0 0.0.0.255 area 0

network 192.168.3.0 0.0.0.255 area 0

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

**router 3:**

hostname R3

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip vrf Google

ip vrf Microsoft

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214420G7

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

vlan 10,20

interface Loopback0

ip vrf forwarding Microsoft

ip address 192.168.4.3 255.255.255.0

interface GigabitEthernet0/0/0

ip vrf forwarding Google

ip address 192.168.4.2 255.255.255.0

negotiation auto

interface GigabitEthernet0/0/1

no ip address

negotiation auto

interface GigabitEthernet0/0/1.10

encapsulation dot1Q 1 native

ip vrf forwarding Google

ip address 192.168.3.30 255.255.255.0

interface GigabitEthernet0/0/1.20

encapsulation dot1Q 2

ip vrf forwarding Microsoft

ip address 192.168.3.31 255.255.255.0

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router ospf 1 vrf Google

network 192.168.3.0 0.0.0.255 area 0

network 192.168.4.0 0.0.0.255 area 0

router ospf 2 vrf Microsoft

network 192.168.3.0 0.0.0.255 area 0

network 192.168.4.0 0.0.0.255 area 0

router ospf 3

router-id 3.3.3.3

network 192.168.3.0 0.0.0.255 area 0

network 192.168.4.0 0.0.0.255 area 0

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

**problems**

During my initial attempt at configuring the network, I faced difficulties due to the topology I had set up. I had created unnecessary subnets, which made the network complex and difficult to manage. I later realized this problem and simplified the network by reducing the number of subnets from 8 to 4. Each subnet was associated with a specific zone in the network, and a horizontal line drawn from each router would intersect each zone.

In my second attempt, I tried to configure the IP addresses before setting up the VRF and assigning ports to their respective groups. This mistake caused an error message indicating that there was an overlap in subnets. I realized my error and rectified it by configuring the VRF and assigning ports to the specified groups before setting up the IP addresses.

During my third attempt, I successfully configured the network with VRF, but I encountered another issue when I found that there were no OSPF routes in the routing table when I used the "Show ip route VRF {vrf-name}" command. I realized that I was missing a critical command and learned about the "Router ospf {as-number} VRF {vrf-name}" command. After implementing this command, all the necessary routes and commands were added to achieve the desired results.

In summary, while configuring the network, I encountered several challenges and made a few mistakes. However, I was able to learn from these mistakes and resolve them. By simplifying the network topology, configuring the VRF and assigning ports correctly, and learning about the necessary OSPF commands, I was able to successfully configure the network.

**Conclusion**

During this lab, I gained valuable knowledge and experience about VRF, including how it is used and implemented. This included learning about the different commands and configurations necessary to create and manage VRF tables, as well as how to assign interfaces to specific VRF groups and display VRF tables using the "show ip route vrf {vrf-name}" command. Additionally, I discovered the importance of configuring OSPF processing for a specific VRF table using the "router ospf {as-number} vrf {vrf-name}" command, which allowed me to successfully create a functioning network.