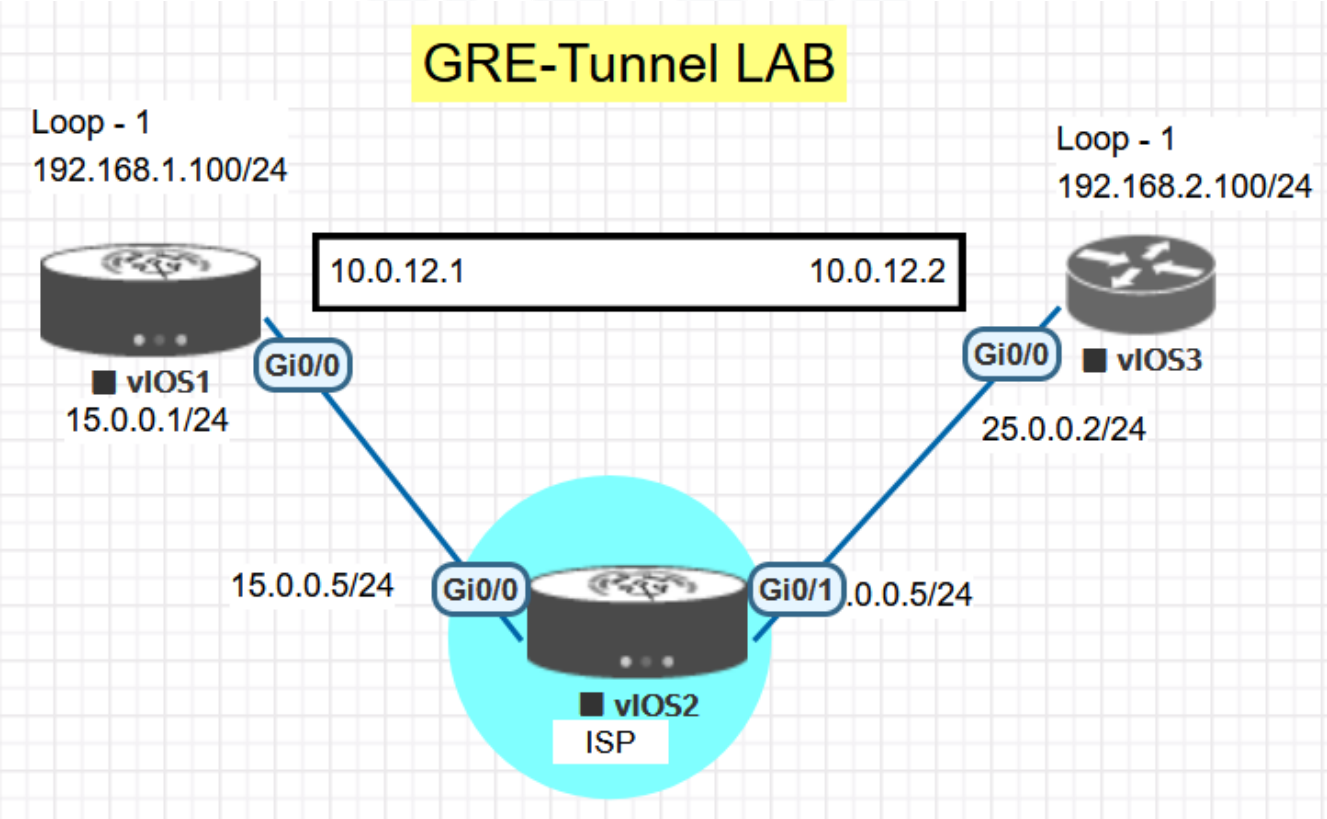
PyNetLabs
Network Automation Specialists

GRE TUNNEL LAB

Generic Routing Encapsulations (GRE) –

It is a tunneling protocol that can be used to encapsulate a wide variety of different network protocols inside a point-to-point link between two endpoints on an IP network. GRE allows networks to send packets across a network encapsulated in another protocol, allowing the network to act as if it's just another network segment.

GRE is developed by CISCO & these GRE tunnels are not encrypted and very easy to configure let's see how?



Create a lab like – using 3 routers and we'll make sure there will be GRE tunnel between the R-1 & R-3 & R-2 is acting as your ISP Router.

- Point-to-Point GRE Tunnel over internet (B/w R-1 & R-3)
- Routing Protocol on tunnel for verification
- Also make sure reachability should be there from R-1 & R-3 between the tunnel source and destination.
- We'll use static routes for that in all the three routers.

Let's configure the lab -

```
R-1>enable
R-1#configure terminal
R-1(config)#hostname R-1
R-1(config)#int gig0/0
R-1(config-if)#ip address 15.0.0.1 255.255.255.0
R-1(config-if)#no shutdown
R-1(config-if)#exit
R-1(config)#interface loopback 1
R-1(config-if)#ip address 192.168.1.100 255.255.255.0
R-1(config-if)#exit

R-2>enable
R-2#configure terminal
R-2(config)#hostname R-2
R-2(config)#int gig0/0
R-2(config-if)#ip address 15.0.0.5 255.255.255.0
R-2(config-if)#no shutdown
R-2(config-if)#exit
R-2(config)#interface gig0/1
R-2(config-if)#ip address 25.0.0.5 255.255.255.0
R-2(config-if)#no shutdown
R-2(config-if)#exit
```

```
R-3>enable
R-3#configure terminal
R-3(config)#hostname R-3
R-3(config)#int gig0/0
R-3(config-if)#ip address 25.0.0.2 255.255.255.0
R-3(config-if)#no shutdown
R-3(config-if)#exit
R-3(config)#interface loop 1
R-3(config-if)#ip address 192.168.2.100 255.255.255.0
R-3(config-if)#exit
```

Now we'll put some static routing in between all the devices for ensuring the reachability.

```
R-1(config)#ip route 25.0.0.0 255.255.255.0 15.0.0.5
R-1(config)#ip route 192.168.2.0 255.255.255.0 15.0.0.5

R-3(config)#ip route 192.168.1.0 255.255.255.0 25.0.0.5
R-3(config)#ip route 15.0.0.0 255.255.255.0 25.0.0.5

R-2(config)#ip route 192.168.1.0 255.255.255.0 15.0.0.1
R-2(config)#ip route 192.168.2.0 255.255.255.0 25.0.0.2
```

Verifications for reachability – (R-3) & (R-1) where we'll create a tunnel in b/w,

```
R-3(config)#do ping 192.168.1.100
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/12/20 ms
```

```
R-1(config)#do ping 192.168.2.100
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/22 ms
```

Now we will create the GRE Tunnel between R-1 & R-3

```
R-1(config)#interface tunnel ?
  <0-2147483647>  Tunnel interface number

R-1(config)#interface tunnel 10
R-1(config-if)#
*May  15  05:23:00.168:  %LINEPROTO-5-UPDOWN:  Line  protocol  on
Interface Tunnel10, changed state to down

R-1(config-if)#ip address 10.0.12.1 255.255.255.0
R-1(config-if)#tunnel source 15.0.0.1
R-1(config-if)#tunnel destination 25.0.0.2
R-1(config-if)#exit
*May  15  05:23:39.821:  %LINEPROTO-5-UPDOWN:  Line  protocol  on
Interface Tunnel10, changed state to up

R-3(config)#interface tunnel 10
R-3(config-if)#
*May  15  05:25:39.240:  %LINEPROTO-5-UPDOWN:  Line  protocol  on
Interface Tunnel10, changed state to down
R-3(config-if)#ip add 10.0.12.2 255.255.255.0
R-3(config-if)#tunnel source 25.0.0.2
R-3(config-if)#tunnel destination 15.0.0.1
R-3(config-if)#exit
*May  15  05:26:15.327:  %LINEPROTO-5-UPDOWN:  Line  protocol  on
Interface Tunnel10, changed state to up
```

Verifications of Tunnel -

```
R-1(config)#do show ip int tunnel 10
Tunnel10 is up, line protocol is up
  Internet address is 10.0.12.1/24
```

```
R-3(config)#do show ip int tunnel 10
Tunnel10 is up, line protocol is up
  Internet address is 10.0.12.2/24
```

PyNetLabs
Network Automation Specialists

Now we'll also implement EIGRP on tunnel to check the connectivity and neighbor table verifications -

```
R-1(config)#router eigrp 100
R-1(config-router)#no auto-sum
R-1(config-router)#network 10.0.12.0
R-1(config-router)#exit

R-3(config)#router eigrp 100
R-3(config-router)#no auto-sum
R-3(config-router)#network 10.0.12.0
R-3(config-router)#exi

*May 15 05:32:23.200: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.0.12.1 (Tunnel10) is up: new adjacency
```

Verify the neighbor table of EIGRP on (R-1 & R-3)

```
R-3(config)#do show ip eigrp neighbor
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   10.0.12.1               Tu10                     14 00:01:23   34  1470  0  1
R-3(config)#
```

```
R-1(config)#do show ip eigrp neighbor
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   10.0.12.2               Tu10                     14 00:01:52   47  1470  0  1
R-1(config)#
```

So, that's the complete lab setup and verifications of GRE Tunnel.