Task 1.1

SW1 and SW2:

```
interface FastEthernet0/7
 spanning-tree portfast
 spanning-tree bpduguard enable
```

Task 1.2

SW1:

```
interface FastEthernet0/21
 switchport trunk allowed vlan 102
```

☞ **Quick Note**
Only VLAN 102 is allowed.

SW2:

```
interface range Fa0/16 - 18
 switchport trunk allowed vlan 1-101,103-4094
```

SW3:

```
interface range Fa0/16 - 20
 switchport trunk allowed vlan 1-101,103-4094
```

☞ **Quick Note**
The **switchport trunk allowed vlan except 102** command will produce the same output in the switch's configuration.

SW4:

```
interface FastEthernet0/15
 switchport trunk allowed vlan 102
!
interface range Fa0/19 - 20
 switchport trunk allowed vlan 1-101,103-4094
```

Task 1.3

SW1:

```
vlan 281
  private-vlan isolated
!
vlan 28
 name VLAN_28
  private-vlan primary
  private-vlan association 281
!
interface FastEthernet0/7
 switchport private-vlan host-association 28 281
 switchport mode private-vlan host
```

SW2:

```
vlan 281
  private-vlan isolated
!
vlan 28
 name VLAN_28
  private-vlan primary
  private-vlan association 281
!
!
```

☞ **Quick Note**
By default devices connected to SW1 port Fa0/7 and SW2 port Fa0/7 will not be able to communicate with SW2's V28 interface.


```
interface FastEthernet0/2
 switchport private-vlan mapping 28 281
 switchport mode private-vlan promiscuous
!
interface FastEthernet0/7
 switchport private-vlan host-association 28 281
 switchport mode private-vlan host
```

Task 1.3 Breakdown

By default all ports within a VLAN have layer 2 reachability between each other. Private VLANs allow for the separation of a single VLAN into multiple segments or sub-broadcast domains by restricting layer 2 communication within the VLAN. A common implementation for Private VLANs would be to restrict communication between web servers within a VLAN but allow access to a DNS server and their default gateway. Although this configuration could be accomplished using protected ports, protected ports only restrict traffic within a single switch. Private VLANs allow for this configuration to span across multiple switches.

Private VLANs require that the switches to be in VTP transparent mode. There are three types of VLANs that make up a private VLAN. The first one is called the primary VLAN. The other two, community and isolated, are referred to as secondary VLANs. Ports that are assigned to an isolated VLAN can not communicate with other ports at layer 2, with the exception of ports in the primary VLAN. Ports assigned within a community can communicate with other ports assigned within the same community, along with ports assigned to the primary VLAN. This means that layer 2 communication is not permitted between two isolated ports, an isolated port and a port within a community, or between two ports within different communities. Also note that these restrictions exclude trunk ports.

There are three types of ports for Private VLANs. The first one is called a promiscuous port. A promiscuous port can communicate via layer 2 to all other promiscuous ports, isolated ports, and community ports. Promiscuous ports are assigned to the primary VLAN. The second port type is called an isolated port. Isolated ports can only communicate via layer 2 to promiscuous ports. The last type is called a community port. A community port can talk to other ports that are within the same community and ports that are promiscuous ports.

Note

Private VLAN Guidelines:

- Private VLANs must be configured in the global configuration; the VLAN database mode configuration is not supported for Private VLANs.
- Private VLAN information is not propagated via VTP.
- Isolated and community VLANs do not run their own instance of spanning tree; if fine-tuning of spanning tree is needed the configuration should be applied to the primary VLAN.
- Although Private VLANs restrict layer 2 communication devices may still be able to communicate if their traffic is routed through a layer 3 device.

Task 1.3 Verification

```
Rack1SW1#show interfaces fa0/7 switchport | include private|28|281
Administrative Mode: private-vlan host
Administrative private-vlan host-association: 28 (VLAN_28) 281
(VLAN0281)
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none

Rack1SW2#show interfaces fa0/2 switchport | include private|28|281
Administrative Mode: private-vlan promiscuous
Operational Mode: private-vlan promiscuous
Administrative private-vlan host-association: none
Administrative private-vlan mapping: 28 (VLAN_28) 281 (VLAN0281)
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan:
  28 (VLAN_28) 281 (VLAN0281)

Rack1SW2#show interfaces fa0/7 switchport | include private|28|281
Administrative Mode: private-vlan host
Administrative private-vlan host-association: 28 (VLAN_28) 281
(VLAN0281)
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
```

For testing purposes we will temporarily change R6's Fa0/0 IP address and VLAN to facilitate the test.

```
Rack1SW2#show running-config interface fa0/6
Building configuration...

Current configuration : 117 bytes
!
interface FastEthernet0/6
 switchport private-vlan host-association 28 281
 switchport mode private-vlan host
end
```



```
Rack1R6#show running-config interface Fa0/0
Building configuration...

Current configuration : 98 bytes
!
interface FastEthernet0/0
 ip address 183.1.28.6 255.255.255.0
end

Rack1R6#ping 183.1.28.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 183.1.28.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Rack1R6#ping 183.1.28.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 183.1.28.8, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Rack1SW2#ping 183.1.28.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 183.1.28.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

Rack1SW2#ping 183.1.28.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 183.1.28.6, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```


Task 2.1

R3:

```
interface Serial1/1
 ip ospf priority 0
```

R4:

```
interface Serial0/0/0
 ip ospf priority 0
!
interface FastEthernet0/0
 ip ospf network non-broadcast
!
router ospf 1
 neighbor 183.1.45.5
```

R5:

```
interface FastEthernet0/1
 ip ospf network non-broadcast
!
router ospf 1
 neighbor 183.1.45.4
```

R6:

```
router ospf 1
 redistribute connected route-map CONNECTED->OSPF subnets
!
ip prefix-list VLAN_6 permit 183.1.6.0/24
!
route-map CONNECTED->OSPF permit 10
 match ip address prefix-list VLAN_6
```

Task 2.1 Verification

Verify the OSPF configuration:

```
Rack1R5#show ip ospf interface
Serial0/0/0 is up, line protocol is up
  Internet Address 183.1.0.5/24, Area 0
  Process ID 1, Router ID 150.1.5.5, Network Type BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 150.1.5.5, Interface address 183.1.0.5
  No backup designated router on this network
<output omitted>
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 150.1.3.3
    Adjacent with neighbor 150.1.4.4

<output omitted>

Loopback0 is up, line protocol is up
  Internet Address 150.1.5.5/24, Area 0
  Process ID 1, Router ID 150.1.5.5, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host
```



```
Rack1R5#show ip ospf neighbor

Neighbor ID   Pri   State        Dead Time   Address         Interface
150.1.3.3     0   FULL/DROTHER   00:00:37    183.1.0.3      Serial0/0/0
150.1.4.4     0   FULL/DROTHER   00:00:38    183.1.0.4      Serial0/0/0

Rack1R4#show ip ospf interface loopback 0
Loopback0 is up, line protocol is up
  Internet Address 150.1.4.4/24, Area 0
  Process ID 1, Router ID 150.1.4.4, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host

Rack1R3#show ip ospf interface loopback 0
Loopback0 is up, line protocol is up
  Internet Address 150.1.3.3/24, Area 0
  Process ID 1, Router ID 150.1.3.3, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host

Rack1R5#show ip route ospf
     150.1.0.0/16 is variably subnetted, 3 subnets, 2 masks
O       150.1.4.4/32 [110/65] via 183.1.0.4, 00:09:06, Serial0/0/0
O       150.1.3.3/32 [110/65] via 183.1.0.3, 00:09:06, Serial0/0/0

Rack1R4#show ip route ospf
     150.1.0.0/16 is variably subnetted, 3 subnets, 2 masks
O       150.1.5.5/32 [110/65] via 183.1.0.5, 00:09:40, Serial0/0/0
O       150.1.3.3/32 [110/65] via 183.1.0.3, 00:09:40, Serial0/0/0
```

Verify the OSPF network types on the segment between R4 and R5

```
Rack1R4#show ip ospf interface FastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 183.1.45.4/24, Area 45
  Process ID 1, Router ID 150.1.4.4,Network Type NON_BROADCAST,Cost: 10
 <output omitted>

Rack1R5#sh ip ospf interface FastEthernet 0/1
FastEthernet0/1 is up, line protocol is up
  Internet Address 183.1.45.5/24, Area 45
  Process ID 1, Router ID 150.1.5.5,Network Type NON_BROADCAST,Cost: 10
<output omitted>

Rack1R5#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address       Interface
150.1.3.3         0   FULL/DROTHER    00:00:37    183.1.0.3     Serial0/0
150.1.4.4         0   FULL/DROTHER    00:00:34    183.1.0.4     Serial0/0
150.1.4.4         1   FULL/BDR        00:01:59    183.1.45.4    FastEthernet0/1
```

Check that VLAN6 prefix is being listed as external:

```
Rack1R4#show ip route ospf
     183.1.0.0/24 is subnetted, 4 subnets
O E2    183.1.6.0 [110/20] via 183.1.46.6, 00:00:10, FastEthernet0/1
<output omitted>
```


Task 2.2

R4:

```
interface FastEthernet0/0
 ip ospf cost 10000
!
router ospf 1
 area 45 virtual-link 150.1.5.5
!
interface Serial0/0/0
 ip ospf dead-interval minimal hello-multiplier 3
```

R5:

```
interface FastEthernet0/1
 ip ospf cost 10000
!
router ospf 1
 area 45 virtual-link 150.1.4.4
!
interface Serial0/0/0
 ip ospf dead-interval minimal hello-multiplier 3
```

Task 2.2 Verification

Verify the OSPF virtual link:

```
Rack1R4#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 150.1.5.5 is up
<output omitted>
  Transit area 45, via interface FastEthernet0/0, Cost of using 10000
<output omitted>
```

Check the OSPF routes:

```
Rack1R4#show ip route ospf
<output omitted>
O       150.1.5.5/32 [110/65] via 183.1.0.5, 00:00:21, Serial0/0/0
O       150.1.3.3/32 [110/65] via 183.1.0.3, 00:00:21, Serial0/0/0
```

Verify the backup:

```
Rack1R4(config)#interface serial 0/0/0
Rack1R4(config-if)#shutdown
Rack1R4(config-if)#do show ip route ospf
<output omitted>
O       183.1.0.0 [110/10064] via 183.1.45.5, 00:00:23, FastEthernet0/0
<output omitted>
O       150.1.5.5/32 [110/10001] via 183.1.45.5, 00:00:23,
FastEthernet0/0
O       150.1.3.3/32 [110/10065] via 183.1.45.5, 00:00:23,
FastEthernet0/0
Rack1R4(config-if)#no shutdown
```

Verify the OSPF timers:

```
Rack1R5#show ip ospf interface S0/0 | include Timer
```



```
  Timer intervals configured, Hello 333 msec, Dead 1, Wait 1,
Retransmit 5

Rack1R4#show ip ospf interface S0/0 | include Timer
  Timer intervals configured, Hello 333 msec, Dead 1, Wait 1,
Retransmit 5

Rack1R3#show ip ospf interface S1/1 | include Timer
  Timer intervals configured, Hello 333 msec, Dead 1, Wait 1,
Retransmit 5
```

Task 2.3

> ☞ **Quick Note**
> Arbitrary metric value. Since the task did not specify a value to be used any value could have been used.

R3:

```
router eigrp 100
 redistribute connected metric 10000 100 255 1 1500 route-map
CONNECTED->EIGRP
!
route-map CONNECTED->EIGRP permit 10
 match interface FastEthernet0/0 FastEthernet0/1
```

R6:

```
key chain EIGRP
 key 1
  key-string CISCO
!
interface Serial0/0
 ip authentication mode eigrp 10 md5
 ip authentication key-chain eigrp 10 EIGRP
```

Task 2.3 Verification

Check that the networks appear as EIGRP external routes:

```
Rack1R1#show ip route eigrp | include D EX
D EX 204.12.1.0/24 [170/2707456] via 183.1.123.2, 00:00:51, Serial0/0/0
D EX    183.1.39.0 [170/2707456] via 183.1.123.2, 00:02:20, Serial0/0/0
```

Check that we have BB1 as EIGRP neighbor with authentication enabled:

```
Rack1R6#show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address      Interface       Hold Uptime   SRTT   RTO  Q  Seq Type
0   54.1.1.254   Se0/0/0           13 00:01:38   70   420  0  91
```

See if we actually receive authenticated packets:

```
Rack1R6#debug eigrp packets hello
<output omitted>
EIGRP: received packet with MD5 authentication, key id = 1
EIGRP: Received HELLO on Serial0/0/0 nbr 54.1.1.254
AS 10, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
```


Task 2.4

SW4:

```
key chain RIP
 key 1
  key-string CISCO
!
interface Vlan102
 ip rip authentication mode md5
 ip rip authentication key-chain RIP
```

Task 2.4 Verification

Check if we have RIP enabled and have the key-chain attached:

```
Rack1SW4#show ip protocols | begin rip
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 14 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface                Send  Recv  Triggered RIP  Key-chain
    Vlan102                  2     2                   RIP
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.10.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.10.1.254         120      00:00:03
  Distance: (default is 120)
```

Check that we are receiving routing information via RIP from BB2:

```
Rack1SW4#show ip route rip
R    222.22.2.0/24 [120/7] via 192.10.1.254, 00:00:22, Vlan102
R    220.20.3.0/24 [120/7] via 192.10.1.254, 00:00:22, Vlan102
R    205.90.31.0/24 [120/7] via 192.10.1.254, 00:00:22, Vlan102
```


Task 2.5

R3:

```
router eigrp 100
 redistribute ospf 1 metric 10000 100 255 1 1500
!
router ospf 1
 redistribute eigrp 100 subnets
!
route-map CONNECTED->EIGRP permit 20
 match interface Serial1/1
```

R5:

```
router eigrp 100
 redistribute ospf 1 metric 10000 100 255 1 1500
!
router ospf 1
 redistribute eigrp 100 subnets
 distance 89 0.0.0.0 255.255.255.255 1
!
access-list 1 permit host 150.1.1.0
```

R6:

```
router eigrp 10
 redistribute ospf 1 metric 10000 100 255 1 1500
!
router ospf 1
 redistribute eigrp 10 subnets
!
route-map CONNECTED->OSPF permit 20
 match interface serial0/0/0
```

SW4:

```
router eigrp 100
 redistribute rip metric 10000 100 255 1 1500
!
router rip
 redistribute eigrp 100 metric 1
```


Task 2.5 Breakdown

◎ Strategy Tip

Take route redistribution step-by-step and verify each step as you go. Example: Redistribute between EIGRP and RIP on SW4. Verify the redistribution by having SW2 ping BB2. If redistribution isn't working properly SW2 would not be able to ping BB2.

The above redistribution section presents three problems based on the current configuration. One of these problems is located on R6, and involves the redistribution of EIGRP into OSPF. In a previous OSPF section on R6 VLAN 6 was advertised into OSPF through redistribution. When this redistribution was configured a route-map was used to limit redistribution to only the VLAN 6 interface. However when EIGRP is then redistributed into OSPF on R6, connected interfaces running EIGRP will not be redistributed into OSPF. This is due to the fact that the route-map *CONNECTED->OSPF* ends in an implicit deny. Therefore either the route-map could be removed from the configuration, or it could be modified to allow the connected Serial interface to be redistributed into OSPF. The same problem occurs on R3 when redistributing into EIGRP.

Since connected redistribution is already occurring with a route-map filter, the Serial1/1 Frame Relay link in the OSPF domain will not be redistributed into EIGRP. This is because the link is treated as a connected interface first before being treated as an OSPF interface. To solve this, like on R6, either the connected to EIGRP route-map could be removed on R3, or it could be modified to include the Serial1/1 link.

The next issue is per the requirement of R5 to route through R3 to get to R1's Loopback interface. R1 advertises its Loopback interface into EIGRP with the network statement. This means that R5 will have this route installed as an EIGRP internal route via SW4 with an administrative distance of 90. Additionally R3 is redistributing this route from EIGRP into OSPF. Therefore R5 will also have this route in the OSPF database as an external route learned from R3, which has an administrative distance of 110. Based on this default behavior R5 will choose the internal EIGRP route due to the lower administrative distance. Therefore to get R5 to route through R3 we can either filter out the advertisement from SW4 to R5, which is not allowed per the requirement, or change the administrative distance.

In the above solution the administrative distance is changed with the statement **`distance 89 0.0.0.0 255.255.255.255 1`**, where 89 is the administrative distance (one lower than EIGRP's 90), 0.0.0.0 255.255.255.255 is the neighbor the route is learned from (any neighbor), and 1 is a standard access-list matching the prefix 150.1.1.0. This means that the distance of the OSPF prefix 150.1.1.0 will be changed to 89, and will therefore be preferred over the EIGRP route.

Task 2.5 Verification

Check that R5 sees 150.1.1.0/24 via OSPF:

```
Rack1R5#show ip route 150.1.1.1
Routing entry for 150.1.1.0/24
  Known via "ospf 1", distance 89, metric 20, type extern 2, forward
metric 64
  Redistributing via eigrp 100
  Advertised by eigrp 100 metric 10000 100 255 1 1500
  Last update from 183.1.0.3 on Serial0/0/0, 00:02:24 ago
  Routing Descriptor Blocks:
  * 183.1.0.3, from 150.1.3.3, 00:02:24 ago, via Serial0/0
      Route metric is 20, traffic share count is 1

Rack1R5#traceroute 150.1.1.1

Type escape sequence to abort.
Tracing the route to 150.1.1.1

  1 183.1.0.3 32 msec 28 msec 32 msec
  2 183.1.123.2 56 msec 56 msec 88 msec
  3 183.1.123.1 32 msec *  32 msec
```

Verify full connectivity with the following TCL script:

```
tclsh
proc ping-internal {} {
  foreach i {
  150.1.1.1
  150.1.2.2
  150.1.3.3
  150.1.4.4
  150.1.5.5
  150.1.6.6
  150.1.7.7
  150.1.8.8
  150.1.10.10
  183.1.0.3
  183.1.0.4
  183.1.0.5
  183.1.123.1
  183.1.123.2
  183.1.123.3
  183.1.17.1
  183.1.17.7
  183.1.28.2
  183.1.28.8
```



```
  183.1.45.4
  183.1.45.5
  183.1.46.4
  183.1.46.6
  183.1.105.5
  183.1.105.10
  183.1.6.6
  183.1.107.7
  183.1.107.10
  192.10.1.10
  204.12.1.3
  54.1.1.6
  } { puts [ exec "ping $i" ] }
}
```

◎ Strategy Tip

By using procedures within TCL it allows you to re-run your ping test without having to paste the foreach loop back into the router. The procedure can be called at any time by just typing the procedure's name on the command line.

Use the following script, to check backbone IGP connectivity:

```
proc ping-external {} {
  foreach i {
  200.0.0.1
  200.0.1.1
  200.0.2.1
  200.0.3.1
  222.22.2.1
  220.20.3.1
  205.90.31.1
  } { puts [ exec "ping $i" ] }
}

Rack1R1#tclsh
Rack1R1(tcl)#proc ping-internal {} {
+>  foreach i {
+>  150.1.1.1
+>  150.1.2.2
+>  150.1.3.3
+>  150.1.4.4
+>  150.1.5.5
+>  150.1.6.6
+>  150.1.7.7
+>  150.1.8.8
+>  150.1.10.10
+>  183.1.0.3
+>  183.1.0.4
+>  183.1.0.5
+>  183.1.123.1
+>  183.1.123.2
```



```
+>  183.1.123.3
+>  183.1.17.1
+>  183.1.17.7
+>  183.1.28.2
+>  183.1.28.8
+>  183.1.39.3
+>  183.1.39.9
+>  183.1.45.4
+>  183.1.45.5
+>  183.1.46.4
+>  183.1.46.6
+>  183.1.105.5
+>  183.1.105.10
+>  183.1.6.6
+>  183.1.107.7
+>  183.1.107.10
+>  192.10.1.10
+>  204.12.1.3
+>  54.1.1.6
+>  } { puts [ exec "ping $i" ] }
+>}

Rack1R1(tcl)#ping-internal

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

<output omitted>

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 54.1.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 140/143/145
ms

Rack1R1(tcl)#ping-external

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 204/205/208
ms

<output omitted>

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 205.90.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 204/205/212
ms

Rack1R1(tcl)#tclquit
Rack1R1#
```


Pitfall

Remember to exit the TCL shell using the **tclquit** command when finished with the reachability verification. If the TCL shell is enabled commands that overlap between TCL and the IOS will be interpreted by TCL and not the IOS. An example is the **set** command used in a route-map. Both TCL and the IOS use the set command. If you try to use the **set** command in a route-map when the TCL shell is still enabled the TCL shell will display an error message:

```
Rack1R1(tcl)#conf t
Rack1R1(config)#route-map TEST
Rack1R1(config-route-map)#set ip next-hop 1.1.1.1
wrong # args: should be "set varName ?newValue?"
Rack1R1(config-route-map)#do tclquit
Rack1R1(config-route-map)#set ip next-hop 1.1.1.1
Rack1R1(config-route-map)#
```

Note

Older Catalyst IOS versions do not support TCL scripting. A smartport macro can be used in place of the TCL shell for ping tests on the switches as follows.

```
Rack1SW3(config)#macro name PINGS
Enter macro commands one per line. End with the character '@'.
do ping 150.1.1.1
do ping 150.1.2.2
<output omitted>
@
Rack1SW3(config)#macro global apply PINGS

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/113/116
ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
<output omitted>
```


Task 2.6

R6:

```
ip as-path access-list 1 permit _54$
!
route-map LOCAL_PREFERENCE permit 10
 match as-path 1
 set local-preference 200
!
route-map LOCAL_PREFERENCE permit 1000
!
router bgp 100
 neighbor 54.1.1.254 route-map LOCAL_PREFERENCE in
```

Task 2.6 Verification

Verify that local preference is correctly set to 200 for routes originating from AS 54:

```
Rack1R6#show ip bgp regexp _54$
<output omitted>
*> 28.119.16.0/24   54.1.1.254                    200      0 54 i
*> 28.119.17.0/24   54.1.1.254                    200      0 54 i
*> 114.0.0.0        54.1.1.254               0    200      0 54 i
*> 115.0.0.0        54.1.1.254               0    200      0 54 i
*> 116.0.0.0        54.1.1.254               0    200      0 54 i
*> 117.0.0.0        54.1.1.254               0    200      0 54 i
*> 118.0.0.0        54.1.1.254               0    200      0 54 i
*> 119.0.0.0        54.1.1.254               0    200      0 54 i
```

And that the other AS paths have a local preference of 100:

```
Rack1R6#show ip bgp regexp _254$
<output omitted>
*>i205.90.31.0      183.1.105.10             0    100      0 200 254 ?
*>i220.20.3.0       183.1.105.10             0    100      0 200 254 ?
*>i222.22.2.0       183.1.105.10             0    100      0 200 254 ?
```

Task 2.7

R1:

```
interface Loopback1
 ip address 150.1.11.1 255.255.255.0
!
router bgp 200
 network 150.1.11.0 mask 255.255.255.0
```

R2:

```
ip prefix-list R1_BGP_LOOPBACK seq 5 permit 150.1.11.0/24
!
route-map MED permit 10
 match ip address prefix-list R1_BGP_LOOPBACK
 set metric 200
!
route-map MED permit 1000
!
router bgp 200
```



```
 neighbor 183.1.123.3 route-map MED out
```

R5:

```
router bgp 100
 neighbor 183.1.0.3 next-hop-self
```

SW4:

```
ip prefix-list R1_BGP_LOOPBACK seq 5 permit 150.1.11.0/24
!
route-map MED permit 10
 match ip address prefix-list R1_BGP_LOOPBACK
 set metric 100
!
route-map MED permit 1000
!
router bgp 200
 neighbor 183.1.105.5 route-map MED out
```

Task 2.7 Verification

Confirm that R3 has two paths to 150.1.11.0/24:

```
Rack1R3#show ip bgp 150.1.11.0
BGP routing table entry for 150.1.11.0/24, version 44
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Flag: 0x820
  Advertised to update-groups:
     1
  200
    183.1.123.1 from 183.1.123.2 (150.1.2.2)
      Origin IGP, metric 200, localpref 100, valid, external
  200
    183.1.0.5 from 183.1.0.5 (150.1.5.5)
      Origin IGP, metric 100, localpref 100, valid, internal, best

Rack1R5#show ip bgp 150.1.11.0
BGP routing table entry for 150.1.11.0/24, version 38
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     2          3
  200
    183.1.105.10 from 183.1.105.10 (150.1.10.10)
      Origin IGP, metric 100, localpref 100, valid, external, best
```

Verify that backup works:

```
Rack1R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R5(config)#interface FastEthernet 0/0
Rack1R5(config-if)#shut

Rack1R5#show ip bgp 150.1.11.0
BGP routing table entry for 150.1.11.0/24, version 29
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     3
  200, (Received from a RR-client)
```



```
    183.1.123.1 (metric 20) from 183.1.0.3 (150.1.3.3)
      Origin IGP, metric 200, localpref 100, valid, internal, best


Rack1R5#traceroute 150.1.11.1

Type escape sequence to abort.
Tracing the route to 150.1.11.1

  1 183.1.0.3 28 msec 32 msec 32 msec
  2 183.1.123.2 44 msec 48 msec 44 msec
  3 183.1.123.1 52 msec *  48 msec
```

Task 3.1

R4:

```
ipv6 unicast-routing
!
interface FastEthernet0/1
 ipv6 address 2001:CC1E:1:404::/64 eui-64
```

R5:

```
ipv6 unicast-routing
!
interface FastEthernet0/0
 ipv6 address 2001:CC1E:1:505::/64 eui-64
```

Task 3.1 Verification

Verify IPv6 addressing:

```
Rack1R5#show ipv6 interface brief
FastEthernet0/0                [up/up]
    FE80::207:EBFF:FEDE:5621
    2001:CC1E:1:505:207:EBFF:FEDE:5621

Rack1R4#show ipv6 interface brief
FastEthernet0/1                [up/up]
    FE80::230:94FF:FE7E:E582
    2001:CC1E:1:404:250:80FF:FE04:8E01
```


Task 3.2

R4:

```
interface Tunnel0
 ipv6 address 2001:CC1E:1:4545::4/64
 tunnel source 150.1.4.4
 tunnel destination 150.1.5.5
 tunnel mode ipv6ip
```

R5:

```
interface Tunnel0
 ipv6 address 2001:CC1E:1:4545::5/64
 tunnel source 150.1.5.5
 tunnel destination 150.1.4.4
 tunnel mode ipv6ip
```

Task 3.2 Verification

Verify the tunnel:

```
Rack1R5#show interfaces tunnel 0
Tunnel0 is up, line protocol is up
<output omitted>
  Tunnel source 150.1.5.5, destination 150.1.4.4
  Tunnel protocol/transport IPv6/IP

Rack1R5#ping 2001:CC1E:1:4545::4

Sending 5, 100-byte ICMP Echos to 2001:CC1E:1:4545::4, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/71/76 ms
```


Task 3.3

R4:

```
ipv6 host Rack1R5 2001:CC1E:1:505:206:D7FF:FEA8:3021
!
interface Tunnel0
 ipv6 rip CISCO enable
!
interface FastEthernet0/1
 ipv6 rip CISCO enable
!
ipv6 router rip CISCO
```

☞ Quick Note
R5's global unicast address based off of EUI-64 host portion.

R5:

```
ipv6 host Rack1R4 2001:CC1E:1:404:250:80FF:FE04:8E01
!
interface Tunnel0
 ipv6 rip CISCO enable
!
interface FastEthernet0/0
 ipv6 rip CISCO enable
!
ipv6 router rip CISCO
```

☞ Quick Note
R4's global unicast address based off of EUI-64 host portion.

Task 3.3 Verification

Check to see that RIPng is configured correctly:

```
Rack1R5#show ipv6 rip
RIP process "CISCO", port 521, multicast-group FF02::9, pid 198
     Administrative distance is 120. Maximum paths is 16
     Updates every 30 seconds, expire after 180
     Holddown lasts 0 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
     Periodic updates 8, trigger updates 1
  Interfaces:
    FastEthernet0/0
    Tunnel0
  Redistribution:
    None

Rack1R4#show ipv6 rip
RIP process "CISCO", port 521, multicast-group FF02::9, pid 192
     Administrative distance is 120. Maximum paths is 16
     Updates every 30 seconds, expire after 180
     Holddown lasts 0 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
     Periodic updates 10, trigger updates 2
  Interfaces:
    FastEthernet0/1
    Tunnel0
  Redistribution:
    None
```

Reachability Verification

```
Rack1R4#show ipv6 route rip
<output omitted>
R   2001:CC1E:1:505::/64 [120/2]
     via FE80::9601:505, Tunnel0

Rack1R4#ping ipv6 Rack1R5

Sending 5, 100-byte ICMP Echos to 2001:CC1E:1:505:207:EBFF:FEDE:5621,
timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/71/72 ms

Rack1R5#ping ipv6 Rack1R4

Sending 5, 100-byte ICMP Echos to 2001:CC1E:1:404:230:94FF:FE7E:E582,
timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/69/72 ms
```


Task 4.1

R4:
```
mpls ip
access-list 1 permit 150.1.0.0 0.0.255.255
!
no mpls ldp advertise-labels
mpls ldp advertise-labels for 1

interface FastEthernet 0/0
 mpls ip
!
interface FastEthernet 0/1
 mpls ip
!
interface Serial 0/0/0
 mpls ip
```

R5:
```
mpls ip
access-list 1 permit 150.1.0.0 0.0.255.255
!
no mpls ldp advertise-labels
mpls ldp advertise-labels for 1

interface FastEthernet 0/0
 mpls ip
!
interface Serial 0/0/0
 mpls ip
```

R6:
```
mpls ip
access-list 1 permit 150.1.0.0 0.0.255.255
!
no mpls ldp advertise-labels
mpls ldp advertise-labels for 1
!
interface FastEthernet 0/0
 mpls ipp
```

Task 4.1 Verification

Check MPLS LDP neighbors on R4, since it peers with both R5 and R6:

```
Rack1R4#show mpls ldp neighbor
    Peer LDP Ident: 150.1.6.6:0; Local LDP Ident 150.1.4.4:0
        TCP connection: 150.1.6.6.52236 - 150.1.4.4.646
        State: Oper; Msgs sent/rcvd: 124/127; Downstream
        Up time: 01:34:08
        LDP discovery sources:
          FastEthernet0/1, Src IP addr: 183.1.46.6
        Addresses bound to peer LDP Ident:
          183.1.46.6      183.1.6.6       54.1.1.6        150.1.6.6
    Peer LDP Ident: 150.1.5.5:0; Local LDP Ident 150.1.4.4:0
```



```
        TCP connection: 150.1.5.5.48660 - 150.1.4.4.646
        State: Oper; Msgs sent/rcvd: 136/157; Downstream
        Up time: 01:34:06
        LDP discovery sources:
          Serial0/0/0, Src IP addr: 183.1.0.5
        Addresses bound to peer LDP Ident:
          183.1.105.5     183.1.45.5      183.1.0.5       150.1.5.5
```

Now check that labels were only generated for Loopback0 interfaces. Notice that all prefixes with except to the 150.1.0.0/24 range don't have labels assigned.

```
Rack1R4#show mpls forwarding-table
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next
Hop
Label  Label or VC   or Tunnel Id      Switched      interface
16     No Label      150.1.3.3/32      0             Se0/0/0
183.1.0.3
17     Pop Label     150.1.5.5/32      11949         Se0/0/0
183.1.0.5
18     Pop Label     150.1.6.6/32      13789         Fa0/1
183.1.46.6

Rack1R5#show mpls forwarding-table
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next
Hop
Label  Label or VC   or Tunnel Id      Switched      interface
16     No Label      150.1.1.0/24      0             Fa0/0
183.1.105.10
17     No Label      150.1.2.0/24      0             Fa0/0
183.1.105.10
18     No Label      150.1.3.3/32      0             Se0/0/0
183.1.0.3
19     Pop Label     150.1.4.4/32      0             Se0/0/0
183.1.0.4
20     18            150.1.6.6/32      0             Se0/0/0
183.1.0.4
21     No Label      150.1.7.0/24      0             Fa0/0
183.1.105.10
22     No Label      150.1.8.0/24      0             Fa0/0
183.1.105.10
23     No Label      150.1.10.0/24     0             Fa0/0
183.1.105.10
24     No Label      183.1.17.0/24     0             Fa0/0
183.1.105.10
25     No Label      183.1.28.0/24     0             Fa0/0
183.1.105.10
26     No Label      183.1.46.0/24     0             Se0/0/0
183.1.0.4
27     No Label      183.1.107.0/24    0             Fa0/0
183.1.105.10
28     No Label      183.1.123.0/24    0             Fa0/0
183.1.105.10

Rack1R6#show mpls forwarding-table
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next
Hop
```



```
Label  Label or VC   or Tunnel Id      Switched      interface
16     16            150.1.3.3/32      0             Fa0/0
183.1.46.4
17     Pop Label     150.1.4.4/32      0             Fa0/0
183.1.46.4
18     17            150.1.5.5/32      0             Fa0/0
183.1.46.4
19     No Label      183.1.0.0/24      0             Fa0/0
183.1.46.4
20     No Label      183.1.45.0/24     0             Fa0/0
183.1.46.4
```

Task 4.2

R5:

```
ip vrf VPN_A
 rd 100:5
 route-target export 100:5
 route-target import 100:6
!
interface Loopback 1
 ip vrf forwarding VPN_A
 ip address 172.16.5.5 255.255.255.0
!
router bgp 100
 address-family vpnv4
  neighbor 150.1.6.6 activate
  neighbor 150.1.6.6 send-community extended
  address-family ipv4 unicast vrf VPN_A
   redistribute connected
```

R6:

```
ip vrf VPN_B
 rd 100:6
 route-target export 100:6
 route-target import 100:5
!
interface Loopback 1
 ip vrf forwarding VPN_B
 ip address 192.168.6.6 255.255.255.0
!
router bgp 100
 address-family vpnv4
  neighbor 150.1.5.5 activate
  neighbor 150.1.5.5 send-community extended
  address-family ipv4 unicast vrf VPN_B
  redistribute connected
```

Task 4.2 Verification

Check BGP peering session:

```
Rack1R6#show bgp vpnv4 unicast all summary
BGP router identifier 150.1.6.6, local AS number 100
BGP table version is 5, main routing table version 5
```



```
3 network entries using 468 bytes of memory
3 path entries using 204 bytes of memory
6/2 BGP path/bestpath attribute entries using 1008 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
2 BGP extended community entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 1 (at peak 1) using 32 bytes of memory
BGP using 1784 total bytes of memory
BGP activity 14/0 prefixes, 20/6 paths, scan interval 15 secs

Neighbor        V          AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd
150.1.5.5       4        100     153     149        5    0    0
00:07:25        1
```

Check that prefixes have been exchanged over BGP:

```
Rack1R6#show bgp vpnv4 unicast all
BGP table version is 5, local router ID is 150.1.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:5
*>i172.16.5.0/24    150.1.5.5                0    100      0 ?
Route Distinguisher: 100:6 (default for vrf VPN_B)
*>i172.16.5.0/24    150.1.5.5                0    100      0 ?
*> 192.168.6.0      0.0.0.0                  0         32768 ?

Rack1R5#show bgp vpnv4 unicast all summary
BGP router identifier 150.1.5.5, local AS number 100
BGP table version is 5, main routing table version 5
3 network entries using 468 bytes of memory
3 path entries using 204 bytes of memory
8/2 BGP path/bestpath attribute entries using 1344 bytes of memory
3 BGP AS-PATH entries using 72 bytes of memory
2 BGP extended community entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 3 (at peak 3) using 96 bytes of memory
BGP using 2232 total bytes of memory
BGP activity 24/0 prefixes, 24/0 paths, scan interval 15 secs

Neighbor        V          AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd
150.1.6.6       4        100     150     155        5    0    0
00:08:47        1
Rack1R5#show bgp vpnv4 unicast all
BGP table version is 5, local router ID is 150.1.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:5 (default for vrf VPN_A)
*> 172.16.5.0/24    0.0.0.0                  0         32768 ?
*>i192.168.6.0      150.1.6.6                0    100      0 ?
Route Distinguisher: 100:6
*>i192.168.6.0      150.1.6.6                0    100      0 ?
```

Check the label stacks for VPN prefixes in R5 and R6:

```
Rack1R6#show ip cef vrf VPN_B 172.16.5.5
172.16.5.0/24
  nexthop 183.1.46.4 FastEthernet0/0 label 17 29

Rack1R5#show ip cef vrf VPN_A 192.168.6.6
192.168.6.0/24
  nexthop 183.1.0.4 Serial0/0/0 label 18 21
```

Do a ping and a traceroute to VPN prefixes:

```
Rack1R5#ping vrf VPN_A 192.168.6.6 source loopback 1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.6.6, timeout is 2 seconds:
Packet sent with a source address of 172.16.5.5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/64 ms

Rack1R5#traceroute vrf VPN_A 192.168.6.6 source loopback 1

Type escape sequence to abort.
Tracing the route to 192.168.6.6

  1 183.1.0.4 [MPLS: Labels 18/21 Exp 0] 64 msec 60 msec 60 msec
  2 192.168.6.6 32 msec *  28 msec
```

Task 5.1

R2:

```
interface Loopback0
 ip pim sparse-mode
!
ip pim send-rp-discovery Loopback0 scope 16
```

R3:

```
interface Loopback0
 ip pim sparse-mode
!
ip pim send-rp-announce Loopback0 scope 16
```

Task 5.1 Verification

Verify that RP mapping information has been disseminated to routers:

```
Rack1R2#show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP-mapping agent (Loopback0)
```



```
Group(s) 224.0.0.0/4
  RP 150.1.3.3 (?), v2v1
    Info source: 150.1.3.3 (?), elected via Auto-RP
         Uptime: 00:03:26, expires: 00:02:31

Rack1R3#show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)

Group(s) 224.0.0.0/4
  RP 150.1.3.3 (?), v2v1
    Info source: 150.1.2.2 (?), elected via Auto-RP
         Uptime: 00:04:03, expires: 00:02:53

Rack1R5#show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 150.1.3.3 (?), v2v1
    Info source: 150.1.2.2 (?), elected via Auto-RP
         Uptime: 00:04:32, expires: 00:02:26
```


Task 5.2

R5:

```
interface FastEthernet0/0
 ip igmp join-group 226.26.26.26
!
ip mroute 0.0.0.0 0.0.0.0 183.1.0.3
```

Task 5.2 Verification

Before the static mroute is configured on R5:

```
Rack1R2#ping
Protocol [ip]:
Target IP address: 226.26.26.26
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Interface [All]: Serial0/0
Time to live [255]:
Source address: 183.1.2.2
…
Rack1R5# debug ip mpacket
IP(0): s=183.1.2.2 (Serial0/0/0) d=226.26.26.26 id=165, ttl=254,
prot=1, len=104(100), not RPF interface
IP(0): s=183.1.2.2 (Serial0/0/0) d=226.26.26.26 id=166, ttl=254,
prot=1, len=104(100), not RPF interface

Rack1R5#sh ip mroute
<output omitted>

(183.1.2.2, 226.26.26.26), 00:00:15/00:02:44, flags: L
  Incoming interface: FastEthernet0/0, RPF nbr 183.1.105.10
  Outgoing interface list:
    Serial0/0/0, Forward/Sparse-Dense, 00:00:16/00:00:00
```

After the static mroute is configured:

```
Rack1R2#ping
Protocol [ip]:
Target IP address: 226.26.26.26
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Interface [All]: Serial0/0
Time to live [255]:
Source address: 183.1.2.2

Reply to request 0 from 183.1.0.5, 64 ms
Reply to request 0 from 183.1.0.5, 192 ms
Reply to request 1 from 183.1.0.5, 60 ms
Reply to request 1 from 183.1.0.5, 188 ms

Rack1R5#sh ip mroute
<output omitted>
```



```
(183.1.2.2, 226.26.26.26), 00:00:15/00:02:59, flags: LJT
  Incoming interface: Serial0/0/0, RPF nbr 183.1.0.3, Mroute
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse-Dense, 00:00:16/00:02:54
```

Task 5.3

R3:

```
access-list 1 deny   239.0.0.0 0.255.255.255
access-list 1 permit any
!
interface FastEthernet0/0
 ip igmp access-group 1
```

Task 5.3 Verification

```
Rack1R3#show ip igmp interface FastEthernet 0/0 | include access
  Inbound IGMP access group is 1

Rack1R3#show ip access-lists 1
Standard IP access list 1
    10 deny   239.0.0.0, wildcard bits 0.255.255.255
    20 permit any (1 match)
```


Task 6.1

R3:

```
ip access-list extended SYN_ATTACK
 permit tcp any host 183.1.28.100 eq www syn log-input
 permit ip any any
!
interface FastEthernet0/0
 ip access-group SYN_ATTACK in
```

SW4:

```
ip access-list extended SYN_ATTACK
 permit tcp any host 183.1.28.100 eq www syn log-input
 permit ip any any
!
interface Vlan102
 ip access-group SYN_ATTACK in
```

Task 6.1 Verification

Generate TCP SYN packets from BB2 and watch the ACL log hits on SW2:

```
BB2>telnet 183.1.28.100 80
Trying 183.1.28.100, 80 ...

Rack1SW2#show logging

<output omitted>
%SEC-6-IPACCESSLOGP: list SYN_ATTACK permitted tcp 192.10.1.254(18518) (Vlan102
0010.7b3a.14cc) -> 183.1.28.100(80), 1 packet
```


Task 6.2

R3:

```
ip access-list extended SYN_ATTACK
 deny ip 183.1.0.0 0.0.255.255 any
 permit tcp any host 183.1.28.100 eq www syn log-input
 permit ip any any
```

SW4:

```
ip access-list extended SYN_ATTACK
 deny ip 183.1.0.0 0.0.255.255 any
 permit tcp any host 183.1.28.100 eq www syn log-input
 permit ip any any
```

R6:

```
ip access-list extended SYN_ATTACK
 deny   ip 183.1.0.0 0.0.255.255 any
 permit ip any any
!
interface Serial0/0/0
 ip access-group SYN_ATTACK in
```

Task 6.2 Verification

```
Rack1R3#sh ip access-lists  | beg SYN_ATTACK
Extended IP access list SYN_ATTACK
    10 deny ip 183.1.0.0 0.0.255.255 any
    20 permit tcp any host 183.1.28.100 eq www syn log-input
    30 permit ip any any (3 matches)

Rack1R6#sh ip access-lists  | beg SYN_ATTACK
Extended IP access list SYN_ATTACK
    10 deny ip 183.1.0.0 0.0.255.255 any
    20 permit ip any any (20 matches)

Rack1SW2#sh ip access-lists  | beg SYN_ATTACK
Extended IP access list SYN_ATTACK
    10 deny ip 183.1.0.0 0.0.255.255 any
    20 permit tcp any host 183.1.28.100 eq www syn log-input
    30 permit ip any any (19 matches)
```


Task 6.3

SW4:

```
interface Vlan102
 no ip unreachables
 no ip mask-reply
```

Task 6.3 Verification

```
Rack1SW4#show ip interface vlan 102
Vlan102 is up, line protocol is up
  Internet address is 192.10.1.10/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are never sent
  ICMP mask replies are never sent
  IP fast switching is disabled
  IP CEF switching is disabled
  IP Null turbo vector
  IP multicast fast switching is disabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

Task 7.1

R2:

```
rmon alarm 1 ifEntry.11.1 60 delta rising-threshold 15000 1 falling-
threshold 5000 2
rmon event 1 trap IETRAP description "Above 15000 for ifInUcastPkts"
rmon event 2 trap IETRAP description "Below 5000 for ifInUcastPkts"
snmp-server host 183.17.1.100 IETRAP
```


Task 7.1 Verification

Verify RMON configuration:

```
Rack1R2#show rmon alarms
Alarm 1 is active, owned by config
 Monitors ifEntry.11.1 every 60 second(s)
 Taking delta samples, last value was 0
 Rising threshold is 15000, assigned to event 1
 Falling threshold is 5000, assigned to event 2
 On startup enable rising or falling alarm

Rack1R2#show rmon events
Event 1 is active, owned by config
 Description is Above 15000 for ifInUcastPkts
 Event firing causes trap to community IETRAP,
 last event fired at  0y0w0d,00:00:00,
 Current uptime       0y0w0d,06:11:00
Event 2 is active, owned by config
 Description is Below 5000 for ifInUcastPkts
 Event firing causes trap to community IETRAP,
 last event fired at  0y0w0d,00:00:00,
 Current uptime       0y0w0d,06:11:00
```

Task 7.2

R3:

```
ntp server 204.12.1.254
ntp peer 150.1.6.6
```

R6:

```
ntp server 54.1.1.254
```

R1, R2, and SW1:

```
ntp server 150.1.3.3
```

R4, R5, and SW4:

```
ntp server 150.1.6.6
```

Task 7.2 Verification

Verify NTP status and associations:

```
Rack1R3#show ntp status
Clock is synchronized, stratum 5, reference is 204.12.1.254
<output omitted>

Rack1R3#show ntp associations

      address     ref clock st  when  poll reach  delay  offset    disp
+~150.1.6.6    54.1.1.254   5    61    64    6    92.7  50583.  15875.
*~204.12.1.254 127.127.7.1  4    35    64  377     7.5   -1.70     0.7
 * master (synced), # master (unsynced), + selected, - candidate, ~
configured

Rack1R3#show ntp associations detail
150.1.6.6 configured, selected, sane, valid, stratum 5
```

```
ref ID 54.1.1.254, time AF67AB02.8F6D2C86 (06:19:46.560 UTC Sat Apr 3
1993)
our mode active, peer mode passive,our poll intvl 64,peer poll intvl 64
<output omitted>

204.12.1.254 configured, our_master, sane, valid, stratum 4
ref ID 127.127.7.1, time AF67AAB6.27A770F0 (06:18:30.154 UTC Sat Apr 3
1993)
our mode client, peer mode server, our poll intvl 64, peer poll intvl
64
<output omitted>

Rack1SW1#show ntp status
Clock is synchronized, stratum 6, reference is 150.1.3.3
<output omitted>

Rack1SW1#show ntp associations

      address     ref clock  st when  poll reach  delay  offset    disp
*~150.1.3.3     204.12.1.254 5    50    64  340    38.1    0.75  16000.
 * master (synced), # master (unsynced), + selected, - candidate, ~
configured
```

Task 7.3

R3:

```
ntp authentication-key 1 md5 CISCO
```

R6:

```
ntp authentication-key 1 md5 CISCO
```

R1, R2, and SW1:

```
ntp authentication-key 1 md5 CISCO
ntp authenticate
ntp trusted-key 1
ntp server 150.1.3.3 key 1
```

R4, R5, and SW4:

```
ntp authentication-key 1 md5 CISCO
ntp authenticate
ntp trusted-key 1
ntp server 150.1.6.6 key 1
```


Task 7.3 Verification

```
Rack1R1#show ntp associations detail
150.1.3.3 configured, authenticated, our_master, sane, valid, stratum 6
ref ID 204.12.1.254, time CCEC61CE.6070F38F (04:06:38.376 UTC Fri Dec 12 2008)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 47.26 msec, root disp 11.40, reach 377, sync dist 74.097
delay 70.27 msec, offset 0.8069 msec, dispersion 3.94
precision 2**18, version 3
org time CCEC6203.7CB0702A (04:07:31.487 UTC Fri Dec 12 2008)
rcv time CCEC6203.8729CADF (04:07:31.527 UTC Fri Dec 12 2008)
xmt time CCEC6203.715D99BE (04:07:31.442 UTC Fri Dec 12 2008)
filtdelay =    84.85   84.67   84.37   84.37   70.27   69.08   69.27   69.96
filtoffset =    1.52    0.88   -0.17   -0.67    0.81    0.86    0.21    0.04
filterror =     0.02    0.99    1.97    2.62    3.60    4.58    5.55    5.57

Rack1R4#show ntp associations detail
150.1.6.6 configured, authenticated, our_master, sane, valid, stratum 5
ref ID 54.1.1.254, time CCEC6217.A1919786 (04:07:51.631 UTC Fri Dec 12 2008)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 29.75 msec, root disp 2.81, reach 377, sync dist 19.302
delay 3.05 msec, offset -1.2642 msec, dispersion 0.09
precision 2**18, version 3
org time CCEC621B.BE0A1D73 (04:07:55.742 UTC Fri Dec 12 2008)
rcv time CCEC621B.BEC170E5 (04:07:55.745 UTC Fri Dec 12 2008)
xmt time CCEC621B.BDE7063D (04:07:55.741 UTC Fri Dec 12 2008)
filtdelay =     3.05    3.08    3.10    3.45    3.17    3.14    3.13    3.23
filtoffset =   -1.26   -1.26   -1.28   -1.03   -0.99   -0.75   -0.23   -0.19
filterror =     0.02    0.99    1.97    2.61    3.59    4.56    5.54    5.55
```

Task 7.4

R2:

```
interface Serial0/0
 ip accounting precedence input
 ip accounting precedence output
!
ip accounting-threshold 50000
```

R3:

```
interface Serial1/0
 ip accounting precedence input
 ip accounting precedence output
!
ip accounting-threshold 50000
```

Task 7.4 Verification

Verify precedence accounting:

```
Rack1R2#show interfaces serial 0/0 precedence
Serial0/0
  Input
    Precedence 6:  114 packets, 8737 bytes
  Output
    Precedence 0:  1 packets, 114 bytes
    Precedence 6:  119 packets, 8051 bytes

Rack1R3#show interfaces serial 1/0 prec
```



```
Serial1/0
  Input
    Precedence 6:  35 packets, 2706 bytes
  Output
    Precedence 0:  1 packets, 114 bytes
    Precedence 6:  98 packets, 6966 bytes
```

Task 7.5

R5:

```
interface FastEthernet0/0
 standby 1 ip 183.1.105.254
 standby 1 preempt
 standby 1 track Serial0/0/0 100
```

SW4:

```
interface FastEthernet0/18
 standby 1 ip 183.1.105.254
 standby 1 priority 50
 standby 1 preempt
```

Task 7.5 Verification

Verify HSRP configuration:

```
Rack1R5#show standby
Ethernet0/0 - Group 1
  State is Active
    2 state changes, last state change 00:01:16
  Virtual IP address is 183.1.105.254
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.896 secs
  Preemption enabled
  Active router is local
  Standby router is 183.1.105.10, priority 50 (expires in 7.892 sec)
  Priority 100 (default 100)
    Track interface Serial0/0/0 state Up decrement 100

Rack1R5(config)#interface Serial 0/0/0
Rack1R5(config-if)#shutdown
<output omitted>
%HSRP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Active -> Speak

Rack1R5(config-if)#do show standby
Ethernet0/0 - Group 1
  State is Standby
  <output omitted>
  Active router is 183.1.105.10, priority 50 (expires in 8.200 sec)
  Standby router is local
  Priority 0 (default 100)
    Track interface Serial0/0/0 state Down decrement 100
  IP redundancy name is "hsrp-Fa0/0-1" (default)
```


Task 7.6

R3:

```
access-list 2 permit 183.1.0.0 0.0.255.255
!
ip nat inside source list 2 interface FastEthernet0/0 overload
!
interface FastEthernet0/0
 ip nat outside
!
interface Serial1/0
 ip nat inside
!
interface Serial1/1
 ip nat inside
```

Task 7.6 Verification

Verify the NAT translations:

```
Rack1R1#ping 204.12.1.254

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 204.12.1.254, timeout is 2 seconds:
!!!!!

Rack1R3#sh ip nat translations
Pro Inside global     Inside local      Outside local      Outside global
icmp 204.12.1.3:3179  183.1.123.1:3179  204.12.1.254:3179  204.12.1.254:3179
icmp 204.12.1.3:3180  183.1.123.1:3180  204.12.1.254:3180  204.12.1.254:3180
icmp 204.12.1.3:3181  183.1.123.1:3181  204.12.1.254:3181  204.12.1.254:3181
icmp 204.12.1.3:3182  183.1.123.1:3182  204.12.1.254:3182  204.12.1.254:3182
icmp 204.12.1.3:3183  183.1.123.1:3183  204.12.1.254:3183  204.12.1.254:3183
```

Task 8.1

R5:

```
map-class frame-relay DLCI_504
 frame-relay cir 512000
 frame-relay bc 25600
 frame-relay be 51200
 frame-relay mincir 384000
 frame-relay adaptive-shaping becn
!
map-class frame-relay DLCI_513
 frame-relay cir 128000
 frame-relay bc 6400
 frame-relay be 0
 frame-relay mincir 96000
 frame-relay adaptive-shaping becn
!
interface Serial0/0/0
 frame-relay traffic-shaping
 frame-relay interface-dlci 504
  class DLCI_504
 frame-relay interface-dlci 513
```



```
  class DLCI_513
```

Task 8.1 Verification

Check the FRTS configuration:

Rack1R5#show traffic-shape

```
Interface   Se0/0/0
       Access Target Byte  Sustain  Excess   Interval  Increment Adapt
VC     List   Rate   Limit bits/int bits/int (ms)      (bytes)   Active
502           56000  875    7000      0        125       875       -
503           56000  875    7000      0        125       875       -
504           512000 9600   25600     51200    50        3200      BECN
513           128000 800    6400      0        50        800       BECN
501           56000  875    7000      0        125       875       -
```

Double-check for more detailed information:

Rack1R5#show frame-relay pvc 504

```
PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE =
Serial0/0
<output omitted>
  Shaping adapts to BECN
  pvc create time 05:50:23, last time pvc status changed 01:50:51
  cir 512000    bc 25600     be 51200     byte limit 9600   interval 50
  mincir 384000    byte increment 3200  Adaptive Shaping BECN
<output omitted>
```

Note Be is set to 0, to disable bursting:

Rack1R5#show frame-relay pvc 513

```
PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 513, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE =
Serial0/0/0
  <output omitted>
  Shaping adapts to BECN
  pvc create time 05:50:56, last time pvc status changed 04:16:14
  cir 128000    bc 6400      be 0         byte limit 800    interval 50
  mincir 96000     byte increment 800   Adaptive Shaping BECN
  <output omitted>
```

Task 8.2

R1:

```
ip cef
!
class-map match-all ICMP
  match protocol icmp
!
policy-map POLICE_ICMP
  class ICMP
```



```
   police cir 128000 bc 4000
!
interface FastEthernet0/0
 service-policy output POLICE_ICMP
```

Task 8.2 Verification

Check policing parameters:

```
Rack1R1#show policy-map interface fastEthernet 0/0
 FastEthernet0/0

  Service-policy output: POLICE_ICMP

    Class-map: ICMP (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol icmp
      police:
          cir 128000 bps, bc 4000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps
```

Task 8.3

R5:

```
ip cef
!
class-map match-all CITRIX
  match protocol citrix
!
class-map match-all VOICE
  match dscp ef
!
policy-map CBWFQ
 class VOICE
  priority 64
 class CITRIX
  bandwidth remaining percent 30
  queue-limit 16
 class class-default
  fair-queue
!
map-class frame-relay DLCI_504
 service-policy output CBWFQ
```


Task 8.3 Verification

```
Rack1R5#show frame-relay pvc 504

PVC Statistics for interface Serial0/0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0

  input pkts 6             output pkts 3            in bytes 204
  out bytes 102            dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                 out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 3         out bcast bytes 102
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  Shaping adapts to BECN
  pvc create time 01:01:30, last time pvc status changed 01:01:10
  cir 512000    bc 25600     be 51200     byte limit 9600   interval 50
  mincir 384000    byte increment 3200  Adaptive Shaping BECN
  pkts 0         bytes 0         pkts delayed 0         bytes delayed 0
  shaping inactive
  traffic shaping drops 0
  service policy CBWFQ
 Serial0/0/0: DLCI 504 -

  Service-policy output: CBWFQ

    Class-map: VOICE (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:  dscp ef (46)
      Queueing
        Strict Priority
        Output Queue: Conversation 40
        Bandwidth 64 (kbps) Burst 1600 (Bytes)
        (pkts matched/bytes matched) 0/0
        (total drops/bytes drops) 0/0

    Class-map: CITRIX (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol citrix
      Queueing
        Output Queue: Conversation 41
        Bandwidth remaining 30 (%)Max Threshold 16 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0

     Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Queueing
        Flow Based Fair Queueing
        Maximum Number of Hashed Queues 32
        (total queued/total drops/no-buffer drops) 0/0/0
  Output queue size 0/max total 600/drops 0
```

