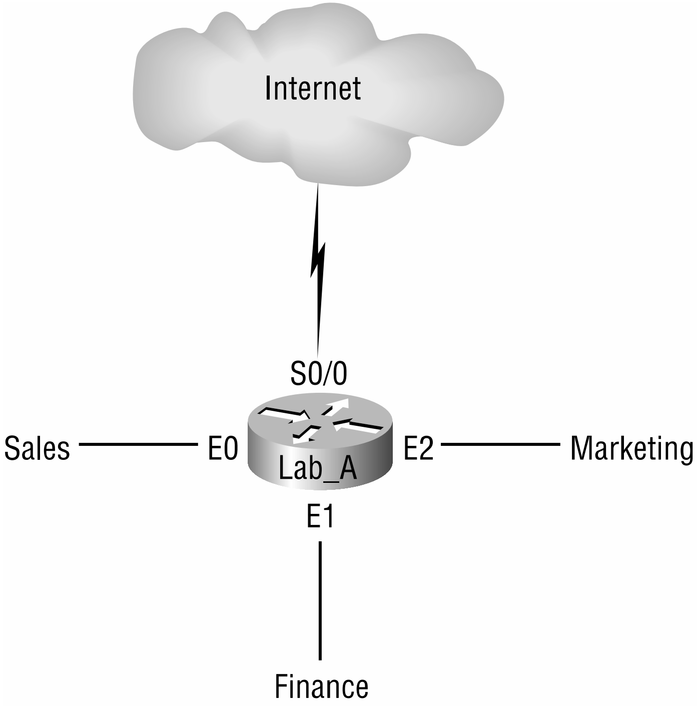
Configuration and Management of Networks

Pedro Amaral

Configuration and Management of Networks

Access Control Lists



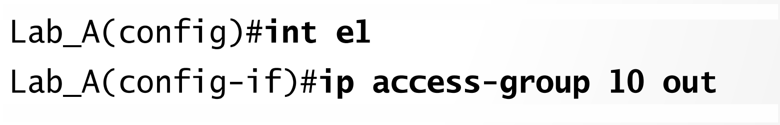
Sales 176.16.40.0/24

Sales cannot have access to finance

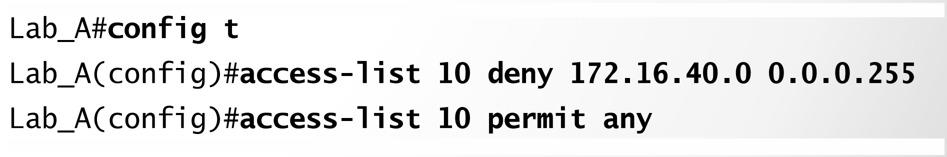
```
nv set acl BLOCK-SALES type ipv4

nv set acl BLOCK-SALES rule 10 match ip source-ip 172.16.40.0/24
nv set acl BLOCK-SALES rule 10 action deny
nv set acl BLOCK-SALES rule 20 action permit

nv set interface swp1 acl BLOCK-SALES outbound
```



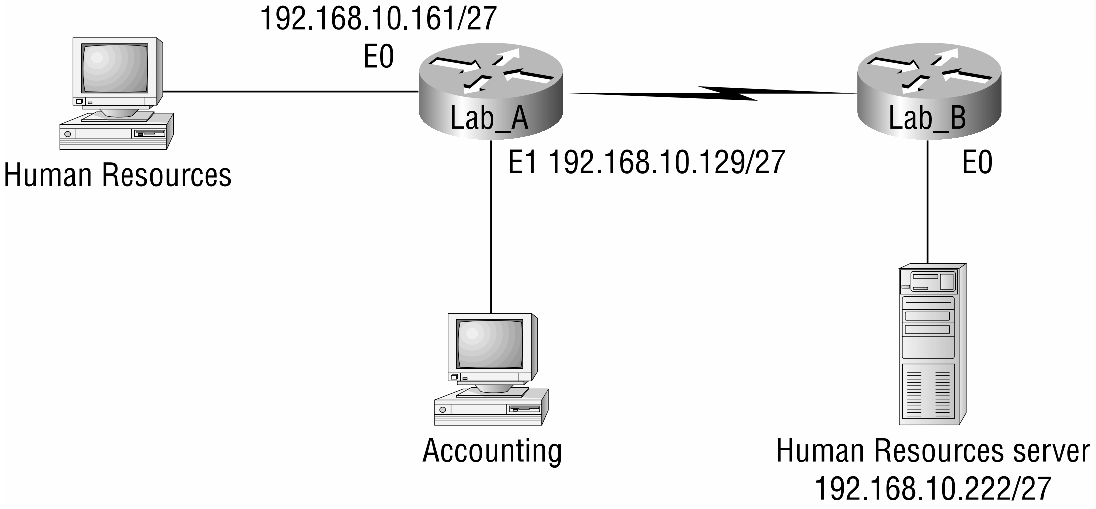
```
Lab_A(config)#int e1
Lab_A(config-if)#ip access-group 10 out
```



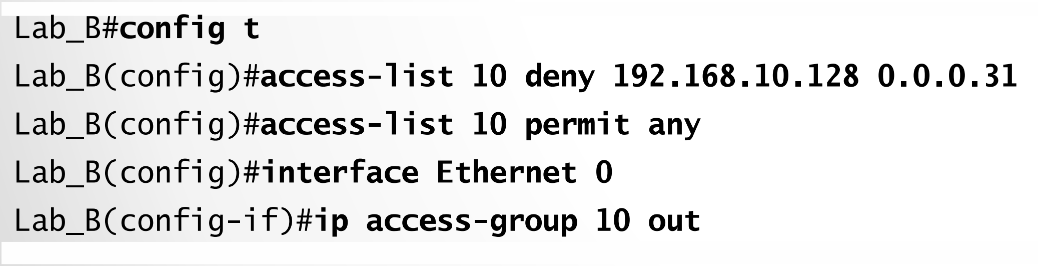
```
Lab_A#config t
Lab_A(config)#access-list 10 deny 172.16.40.0 0.0.0.255
Lab_A(config)#access-list 10 permit any
```

Configuration and Management of Networks

Access Control Lists



Blocks accounting in access to HR server



```
Lab_B#config t
Lab_B(config)#access-list 10 deny 192.168.10.128 0.0.0.31
Lab_B(config)#access-list 10 permit any
Lab_B(config)#interface Ethernet 0
Lab_B(config-if)#ip access-group 10 out
```

```
nv set acl BLOCK-ACCOUNTING type ipv4
nv set acl BLOCK-ACCOUNTING rule 10 match ip source-ip 192.168.10.128/27
nv set acl BLOCK-ACCOUNTING rule 10 action deny
nv set acl BLOCK-ACCOUNTING rule 20 action permit

nv set interface swp1 acl BLOCK-ACCOUNTING outbound
```

Access Control Lists

Only host 172.16.10.3 can ssh to router

```
# Create the ACL
nv set acl SSH-ACCESS type ipv4

# Permit SSH (TCP port 22) from specific host only (rule 10)
nv set acl SSH-ACCESS rule 10 match ip protocol tcp
nv set acl SSH-ACCESS rule 10 match ip source-ip 172.16.10.3/32
nv set acl SSH-ACCESS rule 10 match ip tcp dest-port 22
nv set acl SSH-ACCESS rule 10 action permit

# Deny all other SSH attempts (rule 20)
nv set acl SSH-ACCESS rule 20 match ip protocol tcp
nv set acl SSH-ACCESS rule 20 match ip tcp dest-port 22
nv set acl SSH-ACCESS rule 20 action deny

# Apply to control plane
nv set system control-plane acl SSH-ACCESS inbound
nv config apply
```

IP Prefix Lists

Why Use Prefix Lists Instead of ACLs?

Prefix lists are specifically designed **for filtering routing updates** and offer significant advantages over traditional ACLs :

- Better performance - Optimized for route filtering
- User-friendly syntax - More intuitive than ACL wildcards
- Flexible matching - Match networks with prefix lengths in a specified range

```
# Create a prefix list
nv set router policy prefix-list <NAME> rule <ID> match <PREFIX>

# Optional: Match prefix length range
nv set router policy prefix-list <NAME> rule <ID> match <PREFIX> min-prefix-len <VALUE>
nv set router policy prefix-list <NAME> rule <ID> match <PREFIX> ny>max-prefix-len <VALUE>

# Set action (permit or deny)
nv set router policy prefix-list <NAME> rule <ID> action <permit|de
```

Prefix Lists examples

Example 1: Exact Match (No min/max-prefix-len)

Match only the exact prefix 192.168.0.0/16 - nothing else

```
nv set router policy prefix-list EXACT-MATCH rule 10 match 192.168.0.0/16
nv set router policy prefix-list EXACT-MATCH rule 10 action permit
```

What matches:

✅ 192.168.0.0/16 - Match

❌ 192.168.0.0/20 - No Match (different prefix length)

❌ 192.168.2.0/24 - No Match (different prefix length)

Prefix Lists examples

Example 2: Range Matching with min/max-prefix-len

Match any subnet between /26 and /32 within 10.1.1.0/24

```
nv set router policy prefix-list RANGE-MATCH rule 10 match 10.1.1.0/24
nv set router policy prefix-list RANGE-MATCH rule 10 match 10.1.1.0/24 min-prefix-len 26
nv set router policy prefix-list RANGE-MATCH rule 10 match 10.1.1.0/24 max-prefix-len 32
nv set router policy prefix-list RANGE-MATCH rule 10 action permit
```

What matches:

❌ 10.1.1.0/25 - No Match (prefix length < 26)
✅ 10.1.1.0/27 - Match (26 ≤ 27 ≤ 32)
✅ 10.1.1.64/26 - Match (within 10.1.1.0/24 range, length = 26)
✅ 10.1.1.128/30 - Match (within range)
❌ 10.10.10.0/24 - No Match (different prefix)

TWO conditions **simultaneoulsy**:

-Bits must match for the first 24 bits.

-Prefix length between min (26) and max (32)

Filtering – Route Maps

Route maps are advanced routing policies that combine match criteria with set actions - similar to “if-then”.

Think of route maps as:

- More sophisticated than ACLs or prefix lists
- A scripting language for routing policies
- Combining filtering with route manipulation

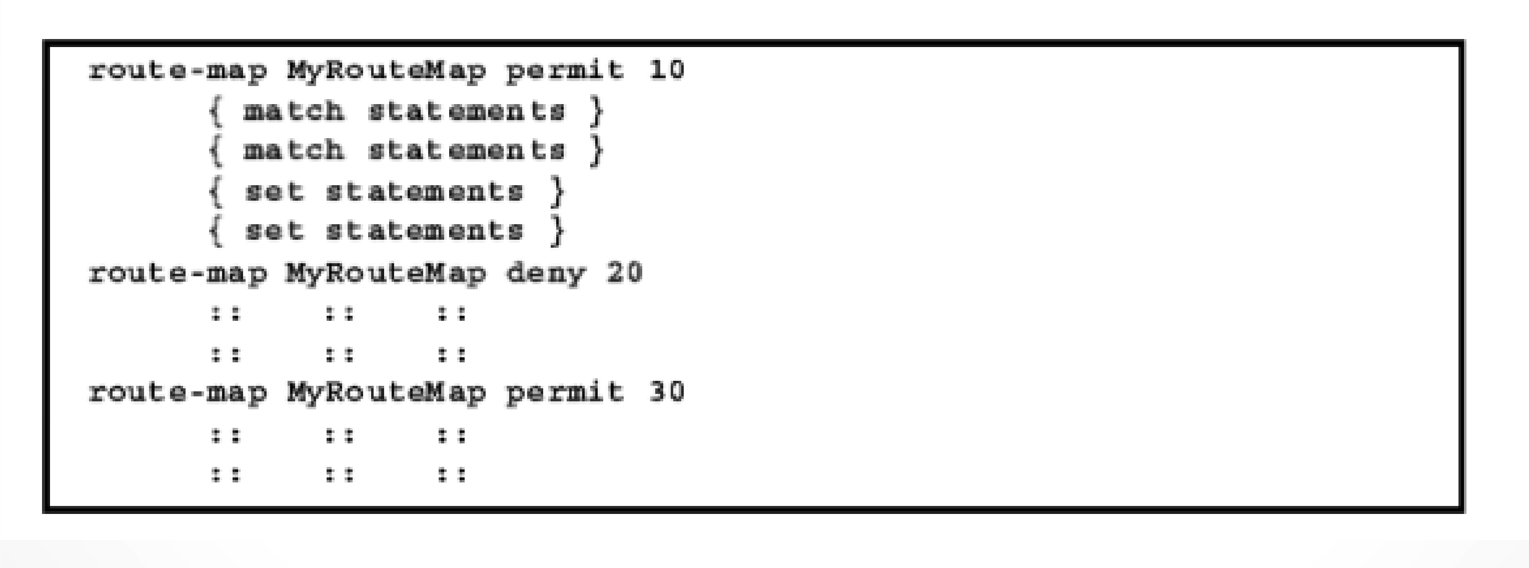
```
# Create route map with rule number
nv set router policy route-map <NAME> rule <ID> action <permit|deny>

# Match criteria (the "IF" part)
nv set router policy route-map <NAME> rule <ID> match <criteria>

# Set actions (the "THEN" part)
nv set router policy route-map <NAME> rule <ID> set <action>

# Apply the configuration
nv config apply
```

Filtering – Route Maps Cisco Sintax



```
route-map MyRouteMap permit 10
      { match statements }
      { match statements }
      { set statements }
      { set statements }
route-map MyRouteMap deny 20
      ::    ::    ::
      ::    ::    ::
route-map MyRouteMap permit 30
      ::    ::    ::
      ::    ::    ::
```

Route Maps

Understanding AND vs. OR Logic

Route maps use conditional logic to determine which routes match and what actions to perform.

Multiple Match Criteria on Separate Lines = AND: All match conditions must be true for the route to match

```
nv set router policy route-map MULTI-AND rule 10 action permit
nv set router policy route-map MULTI-AND rule 10 match interface swp1
nv set router policy route-map MULTI-AND rule 10 match ip-prefix-list MY-PREFIXES
```

Logic: Route **matches IF:**

- Coming from interface swp1 AND
- Matches prefix list MY-PREFIXES

Route Maps

Understanding AND vs. OR Logic

Route maps use conditional logic to determine which routes match and what actions to perform.

Multiple Values in Same Match Type = OR: At least one value must match.

```
# Multiple prefix lists in one match criterion
nv set router policy route-map MULTI-OR rule 10 action permit
nv set router policy route-map MULTI-OR rule 10 match ip-prefix-list LIST1
nv set router policy route-map MULTI-OR rule 10 match ip-prefix-list LIST2
nv set router policy route-map MULTI-OR rule 10 match ip-prefix-list LIST3
```

Logic: Route **matches IF:**

- Matches prefix list LIST1 OR LIST 2 OR LIST 3

Route Maps

Understanding AND vs. OR Logic

Different Rule Sequences = OR: Sequential processing - if a route matches one rule, subsequent rules are not evaluated.

```
nv set router policy route-map BGP-FILTER rule 10 action permit
nv set router policy route-map BGP-FILTER rule 10 match interface swp1

nv set router policy route-map BGP-FILTER rule 20 action permit
nv set router policy route-map BGP-FILTER rule 20 match ip-prefix-list LIST3
```

Logic: Route **matches IF:**

- Coming from interface swp1 OR
- Matches prefix list LIST3

Route Maps

Understanding AND vs. OR Logic

Combined AND+OR

```
nv set router policy route-map COMPLEX rule 10 action permit
nv set router policy route-map COMPLEX rule 10 match as-path-list AS-100
nv set router policy route-map COMPLEX rule 10 match ip-prefix-list LIST-A
nv set router policy route-map COMPLEX rule 10 match ip-prefix-list LIST-B
```

Logic: Route **matches IF:**

- Matches AS-PATH-LIST AS-100 AND
- Matches LIST-A OR LIST-B)

The AS-path must match, plus at least one prefix list

Route Maps

Example: Basic Route Map Structure

Scenario: Create a route map that matches a prefix list and sets an outbound interface

```
# Step 1: Create the prefix list
nv set router policy prefix-list MyList rule 10 match 10.0.0.0/8
nv set router policy prefix-list MyList rule 10 action permit

# Step 2: Create the route map
nv set router policy route-map MyRouteMap rule 10 action permit

# Step 3: Match the prefix list
nv set router policy route-map MyRouteMap rule 10 match ip-prefix-list MyList

# Step 4: Set the outbound interface (for policy-based routing)
nv set router policy route-map MyRouteMap rule 10 set interface swp1

# Apply configuration
nv config apply
```

Route Maps – BGP example

Scenario: Set different MED values for specific networks vs. all other networks when advertising to BGP neighbor.

```
# Step 1: Create prefix list for specific networks
nv set router policy prefix-list PRIORITY-NETS rule 10 match 192.168.25.0/24
nv set router policy prefix-list PRIORITY-NETS rule 10 action permit
nv set router policy prefix-list PRIORITY-NETS rule 20 match 192.168.26.0/24
nv set router policy prefix-list PRIORITY-NETS rule 20 action permit

# Step 2: Create route map with multiple rules
nv set router policy route-map MED-65020 rule 10 action permit
nv set router policy route-map MED-65020 rule 10 match ip-prefix-list PRIORITY-NETS
nv set router policy route-map MED-65020 rule 10 set metric 100

# Step 3: Set default MED for all other routes
nv set router policy route-map MED-65020 rule 100 action permit
nv set router policy route-map MED-65020 rule 100 set metric 200

# Step 4: Apply route map to BGP neighbor (outbound)
nv set vrf default router bgp neighbor 192.168.28.1 address-family ipv4-unicast policy outbound route-map MED-65020

# Step 5: Apply configuration
nv config apply
```