Cisco IOS Commands for CCNA

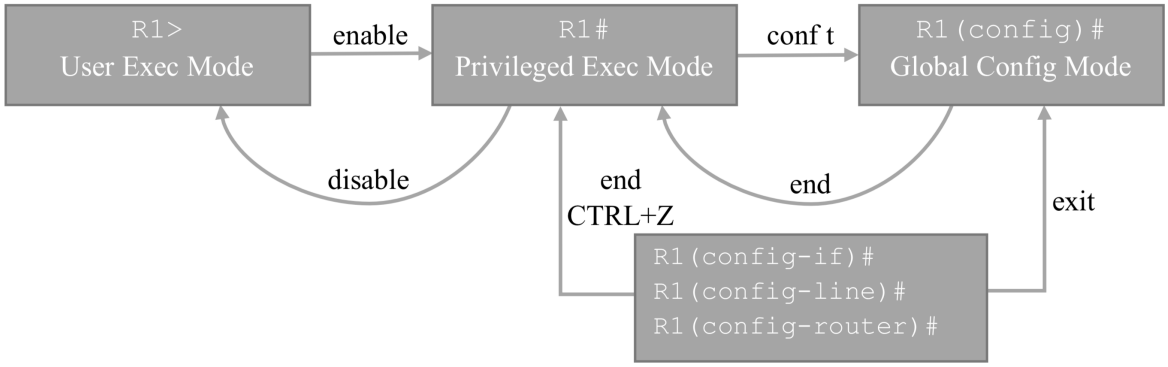
Kaager [†]

August 23, 2018

Contents

[†] https://github.com/Kaager/cisci

1 IOS Modes and Navigation



When describing the use of commands, I generally use the conventions shown in Table 1

`Normal/`**`Bold`**	Normal/**Bold** text indicates commands and keywords that you enter literally as shown.
`Italics`	Italic text indicates arguments for which you supply values.
`[x]`	Square brackets indicate an optional element (keyword or argument).
`{x}`	Braces indiacte a required element (keyword or argument).
`[x{y|z}]`	Braces and vertical lines within square brackets indicate a required choice within an optional element.

Table 1:

2 Basic Setup

2.1 Hostname

```
(config)# hostname hostname
(config)# hostname no hostname[1]
```

2.2 User Exec Password

```
(config)# line console 0[2]
(config-line)# password password
(config-line)# login[3]
(config-line)# exit
```

2.3 Privileged Exec Password

```
(config)# enable secret secret_password
(config)# exit
```

[1] Removes the hostname.
[2] 0 indicates the first, and properly only console port.
[3] Enables user EXEC access

2.4 VTY Line Password

```
(config)# line vty 0 15
(config-line)# password password
(config-line)# login[4]
```

2.5 Encrypt Passwords

```
(config)# service password-encryption
```

2.6 SVI (Switch Virtual Interface) Configuration

```
(config)# interface vlan vlan_id
(config-if)# ip address ip_address subnetmask
(config-if)# no shutdown
```

2.7 Default Gateway on a Switch

```
(config)# ip default-gateway gateway_address
```

2.8 Description on an interface

```
(config-if)# description desired_description
```

2.9 Enable Router to forward IPv6 packets

```
(config)# ipv6 unicast-routing
```

2.10 IPv6 link-local

```
(config-if)# ipv6 address ipv6_address link-local
```

2.11 Banner Message

```
(config)# banner motd # hello, this is the message.  Choose a
character that will start and end the message text, in this
case the hashtag #
```

2.12 Save running config

```
# Copy running-config startup-config
```

3 Miscellaneous

3.1 Stop dem Lookups (Disable DNS Lookup)

```
(config)# no ip domain-lookup
```

[4] Enables VTY access

3.2 If you fucked up

```
# erase startup-config
# delete vlan.dat[5]
# reload
```

3.3 Disable CPD (Cisco Descovery Protocol)

```
(config)# no cdp run
```

3.4 Logging Synchronous

```
(config)# line console 0
(config-line)# logging synchronous
```

3.5 Set Console Time to Never Timeout

```
(config)# line console 0
(config-line)# exec-timeout 0 0[6]
```

4 Backing up and restoring TFTP (Trivial File Transfer Protocol)

```
# copy running-config/startup-config tftp
Enter IP address of the target server:
Enter desired name for the file:
Press 'Enter' to confirm
```

```
# copy tftp running-config/startup-config
Enter the IP address of the host where the configuration file is
stored:
Enter the name to assign to the configuration file:
Press 'Enter' to confirm
```

5 Configuring SSH

The device needs an unique hostname

```
(config)# ip domain-name name.something
(config)# crypto key generate rsa[7]
(config)# ip ssh version 2
(config)# username name secret secret_password
```

```
(config)# ip ssh time-out 60
(config)# ip ssh authentication-retries 2
```

[5] If present, check with "dir flash:"
[6] first 0 is for minutes, second is for seconds
[7] Enter desired key length (1024)

```
(config)# line vty 0 15
(config-line)# login local[8]
(config-line)# transport input ssh[9]
```

5.1 Delete RSA key pair and disables SSH server

```
(config)# crypto key zerosize rsa
```

6 VLAN

6.1 Configure more than 1 interface

```
(config)# interface range type first-number - last-number
```

6.2 Create VLAN

```
(config)# vlan vlan_id
(config-vlan)# name vlan_name
(config-vlan)# end/exit
```

6.3 Assign Ports to VLANs

```
(config)# interface interface_id
(config-if)# switchport mode access
(config-if)# switchport access vlan vlan_id
(config-if)# end/exit
```

Change Port to VLAN 1

```
(config-if)# no switchport access vlan
```

6.4 Delete VLAN

```
(config)# no vlan vlan_id
```

6.5 Trunk Configuration

```
(config)# inteface interface_id
~~(config-if)# switchport trunk encapsulation dot1q~~[10]
(config-if)# switchport mode trunk
```

Specify a Native VLAN Other Than VLAN 1

```
(config-if)# switchport trunk native vlan vlan_id
```

[8] Remote login now looks for local user for authentication
[9] Only able to remote via SSH
[10] Old Command

Specify VLANs Allowed On The Trunk Link

```
(config-if)# switchport trunk allowed vlan vlan_list
```

6.6 Restting Configured Values on the Trunk Links

Set Trunk To Allow All VLANs

```
(config)# inteface interface_id
(config-if)# no switchport trunk allowed vlan
```

Reset Native VLAN to Default

```
(config)# inteface interface_id
(config-if)# no switchport trunk native vlan
```

6.7 To enable trunking from a Cisco switch to a device that does not support DTP (Dynamic Trunking Protocol)

```
(config-if)# switchport mode trunk
(config-if)# switchport nonegotiate
```

6.8 Private VLAN (PVLAN)

Enable Protected Port

```
(config-if)# switchport protected
```

Disable Protected Port

```
(config-if)# no switchport protected
```

7 Inter VLAN Routing

7.1 Router-on-a-Stick

~~Start by enabling Trunking on the switchport connected to the router. Disable DTP.~~

Create a subinterface for a VLAN

```
(config)# interface interface_id.subinterface_id
```

Assign subinterface to a VLAN

```
(config-subif)# encapsulation dot1q vlan_id [native]
```

Assign IP address range

```
(config-subif)# ip address ip_addres subnet_mask
```

When all the subinterfaces have been configured, turn on the physical port

```
(config)# interface interface_id
```

```
(config-if)# no shutdown
```

8 Switching Security

8.1 DHCP Snooping

Enable globally

```
(config)# ip dhcp snooping
```

Enable on specific VLAN

```
(config)# ip dhcp vlan number1, number2
```

Trust a specific interface

```
(config)# interface type interfaceidnumber
(config-if)# ip dhcp trust
```

(Optional) Limit the rate at which an attacker can continually send bogus DHCP requests through untrusted ports to the DHCP server

```
(config-if)# ip dhcp snooping limit rate rate
```

8.2 Port Security

First we have to convert the port(s) to an access port(s), otherwise we can't do port security on it

```
(config-if)# switchport mode access
```

Limit the number of mac-addresses on the port – default is 1

```
(config-if)# switchport port-security maximum number
```

Set violation type (look at Table 2) - default is shutdown

```
(config-if)# switchport port-security violation { protect |
restrict | shutdown }
```

Turn on sticky mac addresses (dynamically learned and added to the running config)

```
(config-if)# switchport port-security mac-address sticky [mac-adr]
```

Lastly turn on port security

```
(config-if)# switchport port-security
```

Violation Mode	Forwards Traffic	Sends Syslog Message	Displays Error Messages	Increases Violation Counter	Shuts Down Port
Protect	No	No	No	No	No
Restrict	No	Yes	No	Yes	No
Shutdown	No	No	No	Yes	Yes

Table 2: Violation Types

9 Show and Verify – VLAN

9.1 Trunk Configuration

```
# show interfaces interface_id switchport
# show interfaces trunk
```

9.2 DTP Mode

```
# show dtp interface interface_id
```

9.3 Troubleshooting Trunks

```
# show interfaces interface_id trunk
```

9.4 Missing VLANs

Is port in correct VLAN

```
# show vlan
# show mac address-table
```

VLAN present in VLAN database

```
# show vlan
# show interfaces
# show interfaces switchport
```

9.5 Port Security Settings

```
# show port-security interface interface_id
```

9.6 Secure MAC Addresses

```
# show port-security address
```

9.7 Routing Table

```
# show ip route
# show ip route static
# show ip route network

# show ipv6 route
# show ipv6 route static
# show ipv6 route network
```

10 Static Routing

```
(config)# ip route network_address subnet_mask {ip_address |
interface_type interface_number [ip_address]} [distance]
[name name] [permamnent] [tag tag]
```

10.1 IPv4

Next-Hop Static Route (Recursive Static Route????)

```
(config)# ip route target_network subnet_mask next_hop_ip_addr
```

Directly Attached/Connected Static Route

```
(config)# ip route target_network subnet_mask exit_int
```

Fully Specified Static Route

```
(config)# ip route target_network subnet_mask exit_int next_
hop_ip_addr
```

Default Static Route

```
(config)# ip route 0.0.0.0 0.0.0.0 {exit_int | next_hop_ip}
```

NOTE: An IPv4 default static route is commonly referred to as a quad-zero.

10.2 IPv6

NOTE: IPv6 routing must be enabled with `(config)# ipv6 unicas- routing`

```
(config)# ipv6 route ipv6-prefix/prefix-lenght {ipv6-addr |
exit-int}
```

Next-Hop Static Route

```
(config)# ipv6 route ipv6-prefix/prefix-lenght next-hop-ipv6-addr
```

Directly Attached/Connected Static Route

```
(config)# ipv6 route ipv6-prefix/prefix-lenght exit-int
```

Fully Specified Static Route

```
(config)# ipv6 route ipv6-prefix/prefix-lenght ipv6-addr exit-int
```

Default Static Route

```
(config)# ipv6 route ::/0 {ipv6-addr | exit-int}
```

10.3 Summary Route IPv4

First we need to calculate the summary route, which can be done in 3 steps:
Step 1: List the networks in binary format.
Step 2: Count the number of far-left matching bits to determine the mask for the summary route. This is the prefix, or subnet mask for the summarized route i.e. /14 or 255.252.0.0
Step 3: Copy the matching bits and add zero-bits to determine the summarized network address.

Example

Step 1:

```
172.20.0.0    10101100.00010100.00000000.00000000
172.21.0.0    10101100.00010101.00000000.00000000
172.22.0.0    10101100.00010110.00000000.00000000
172.23.0.0    10101100.00010111.00000000.00000000
```

Step 2:

```
172.20.0.0    10101100.00010100.00000000.00000000
172.21.0.0    10101100.00010101.00000000.00000000
172.22.0.0    10101100.00010110.00000000.00000000
172.23.0.0    10101100.00010111.00000000.00000000
```

14 matching bits = /14 or 255.252.0.0
Step 3:
`10101100.000101` = matching bits
Add zero-bits:
`10101100.00010100.00000000.00000000`
Answer = 172.20.0.0
Enter the summary route as you would a normal static route, with the summary route as the target network:

```
(config)# ip route 172.20.0.0 255.252.0.0 {exit-int | next-hop}
```

10.4 Summary Route IPv6

Summarizing IPv6 networks into a single IPv6 prefix and prefix-length can be done in seven steps:
Step 1. List the network addresses (prefixes) and identify the part where the addresses differ.
Step 2. Expand the IPv6 if it is abbreviated.
Step 3. Convert the differing section from hex to binary.
Step 4. Count the number of far left matching bits to determine the prefix-length for the summary route. **Note:** The figure shows all networks converted to binary to demonstrate the matching bits; but the summary can also be determined by converting just the lowest and highest network addresses to binary to find the matching bits.
Step 5. Copy the matching bits and then add zero bits to determine the summarized network address (prefix).

Step 6. Convert the binary section back to hex.
Step 7. Append the prefix of the summary route (result of Step 4).

10.5 Floating Static Route

This is a default static route that is configured with the `[distance]` sat higher than 1 (default), with another default static route with a distance of 1. Because the value is greater than 1, the route floats and is not present in the routing table, inless the preferred route fails.

11 RIP - Routing Information Protocol

12 Single Area OSPF

12.1 IPv4

Enter and enable OSPF with

```
(config)# router ospf process_id
```

The *process-id* value represents a number between 1 and 65,535 and is selected by the network administrator. The *process-id* value is locally significant, which means that it does not have to be the same value on the other OSPF routers to establish adjacencies with those neighbors.

Give the router an ID

```
(config-router)# router-id rid
```

The rid value is any 32-bit value expressed as an IPv4 address, e.g. `(config-router)# router-id 1.1.1.1`

If the router already has a name, you might have to clear the OSPF process before you can change it

```
(config-router)# router-id 1.1.1.1

%%% Some error message

# clear ip ospf process
```

Press y to the prompt

```
show ip protocols | section Router ID -> to see the change
```

Using a Loopback Interface as the Router ID

```
(config)# interface loopback 0
(config-if)# ip address 1.1.1.1 255.255.255.255
(config-if)# end
```

The IPv4 address of the loopback interface should be configured using a 32-bit subnet mask

(255.255.255.255). This effectively creates a host route. A 32-bit host route does not get advertised as a route to other OSPF routers. The example displays how to configure a loopback interface with a host route on R1. R1 uses the host route as its router ID, assuming there is no router ID explicitly configured or previously learned.

Enabling OSPF on interfaces

The `network` command determines which interfaces participate in the routing process for an OSPF area. Any interfaces on the router that match the network address in the `network` command are enabled to send and receive OSPF packets. As a result, the network (or subnet) addresses for the interface is included in OSPF routing updates. The basic command syntax is

```
(config-router)#network network_address wildcard_mask area area_id
```

The **area** *area_id* syntax refers to the OSPF area. When configuring single-area OSPF, the **network** command must be configured with the same `area_id` value on all routers. Although any area ID can be used, it is good practice to use an area ID of 0 with single-area OSPF. This convention makes it easier if the network is later altered to support multiarea OSPF.

```
(config)# router ospf 10
(config-router)# network 172.16.1.0 0.0.0.255 area 0
(config-router)# network 172.16.3.0 0.0.0.3 area 0
(config-router)# network 192.168.10.4 0.0.0.3 area 0
```

As an alternative, OSPFv2 can be enabled using the **network** *int_ip_address* **0.0.0.0 area** *area_id* router configuration mode command.
Below you can see an example of specifying the interface IPv4 address with a quad 0 wildcard mask. Entering `network 172.16.3.1 0.0.0.0 area 0` on R1 tells the router to enable interface Serial0/0/0 for the routing process. As a result, the OSPFv2 process will advertise the network that is on this interface (172.16.3.0/30).
The advantage of specifying the interface is that the wildcard mask calculation is not necessary. OSPFv2 uses the interface address and subnet mask to determine the network to advertise.
Note: This method is not really used by Cisco.

```
(config)# router ospf 10
(config-router)# network 172.16.1.1 0.0.0.0 area 0
(config-router)# network 172.16.3.1 0.0.0.0 area 0
(config-router)# network 192.168.10.5 0.0.0.0 area 0
```

Propagate the default route

```
(config-router)# default-information originate
```

Passive Interface

```
(config-router)# passive-interface interface_id
```

Adjusting the interface bandwidth

```
(config)# interface interface_id
(config-if)# bandwidth kilobits
```

Manually setting the OSPF cost

```
(config)# interface interface_id
(config-if)# ip ospf cost value
```

Some show commands

```
show ip ospf neighbor
show ip protocols
show ip ospf
show ip ospf interface
show ip ospf interface brief
show ip ospf interface serial 0/0/1
```

12.2 IPv6

Enable IPv6 routing

```
(config)# ipv6 unicast-routing
```

Router ID

```
(config)# ipv6 router ospf process_id
(config-rtr)# router-id rid
```

Enable OSPFv3 on the interfaces that contain the networks that you want to be part of the OSPF area

```
(config-if)# ipv6 ospf process_id area area_id
```

Prevent routing updates to be send across an interface

```
(config)# ipv6 router ospf process_id
(config-rtr)# passive-interface interface_id
```

Propagate the default route

```
(config-rtr)# redistribute static
```

Some show commands

```
show ipv6 ospf neighbour
show ipv6 protocols
show ipv6 ospf
show ipv6 ospf interface
show ipv6 ospf interface brief
show ipv6 ospf interface serial 0/0/1
show ipv6 route ospf
```

13 EIGRP

13.1 Protocol Configuration

`(config)#` **`router eigrp`** *asn*[11]

Set the Router ID

`(config-router)#` **`eigrp router-id`** *ipv4_address*[12]

Add networks to advertise

`(config-router)#` **`network`** *network_address* [*wildcard_mask*]

Designate passive interfaces

`(config-router)#` **`passive-interface`** {*interface_id* | **`default`**[13]}

Propagate a default static route

`(config-router)#` **`redistribute static`**

Disable automatic route summarization

`(config-router)#` **`no auto-summary`**

Configure K values to manipulate metric formula

`(config-router)#` **`metric weights 0`** *k1 k2 k3 k4 k5*

13.2 Interface Configuration

`(config)#` **`interface`** *interface_id*

Configure the bandwidth parameter

`(config-if)#` **`bandwidth`** *kilobits_bandwidth_value*[14]

Configure EIGRP Manual Summarization[15]

[11] The *asn* (autonomous system number) argument can be assigned to any 16-bit value between the number 1 and 65,535. All routers within the EIGRP routing domain must use the same autonomous system number.

[12] The IPv4 address used to indicate the router ID is actually any 32-bit number displayed in dotted-decimal notation, excluding 0.0.0.0 and 255.255.255.255.

[13] The *default* keyword sets all interfaces as passive.

[14] Default value is 1544 kb/s.

[15] Configure the summary route on all interfaces that send EIGRP packets.

```
(config-if)# ip summary-address eigrp asn network_address
subnet_mask
```

Set maximum bandwidth EIGRP can consume

```
(config-if)# ip bandwidth-percent eigrp asn percentage
```

Configure hello and hold timers

```
(config)# ip hello-interval eigrp asn seconds
(config)# ip hold-time eigrp asn seconds
```

13.3 MD5 Authentication

```
(config)# key chain name_of_chain
(config-keychain)# key key_id[16]
(config-keychain-key)# key-string key_string_text[17]
```

Enable MD5 on an interface

```
(config)# interface interface_id
(config-if)# ip authentication mode eigrp asn md5
(config-if)# ip authentication key-chain eigrp asn name_of_chain
```

13.4 Troubleshooting

```
show ip eigrp interfaces
show ip eigrp neighbors
show ip eigrp topology
show ip eigrp traffic
clear ip eigrp neighbors
```

14 Point-to-Point Connections

14.1 HDLC - High-Level Data Link Control

Cisco uses HDLC as the default encapsulation method on synchronous serial lines. But if it has been changed, change it back with:

```
(config-if)# encapsulation hdlc
```

14.2 PPP - Point-to-Point Protocol

Enable PPP on an interface

```
(config-if)# encapsulation ppp
```

[16] The key ID is the number used to identify an authentication key within a keychain. The range of keys is from 0 to 2,147,483,647. It is recommended that the key number be the same on all routers in the configuration.

[17] The key string is similar to a password. Routers exchanging authentication keys must be configured using the same key string.

PAP

For PAP to work, you need a user on the router

```
(config)# username user_XX secret passw_XX
(config)# interface interface_id
(config-if)# ppp authentication pap
(config-if)# ppp pap sent-username user_YY password passw_YY
```

Mirror these settings at the other end of the link

```
(config)# username user_YY secret passw_YY
(config)# interface interface_id
(config-if)# ppp authentication pap
(config-if)# ppp pap sent-username user_XX password passw_XX
```

CHAP

The hostname on one router must match the username the other router has configured. The password on the 2 users must also match.

```
(config)# username username password password
(config)# interface interface_id
(config-if)# ppp authentication chap
```

14.3 Troubleshoot

Clock rate

The DCE end has to provide the clock signal, look for it under interface serial in the running config.

```
# show running-config | begin interface Serial
```

If it has not been set, set it like this:

```
(config-if)# clock rate clock_rate
```

Authentication

Use the same command as above, look for encapsulation and authentication.
Use the debug command to find further errors.

```
# debug ppp authentication
```

Disbale all debugging with

```
# undebug all
```

Some other shit

Find users

```
# show run | include user
```

```
show interfaces
```

```
show interfaces serial
show ppp multilink
```

15 GRE - Generic Routing Encapsulation

```
(config)# interface tunnel tunnel_id_number
(config-if)# tunnel source ip_address
(config-if)# tunnel destination ip_address
(config-if)# ip address ip_address mask[18]
[(config-if)# tunnel mode gre ip]
```

15.1 Troubleshooting

```
show ip interface brief | include Tunnel
show interface tunnel tunnel_id_number
show ip ospf neighbor
```

16 NTP

[Optional] Change timezone to CET
`(config)# clock timezone CET +1`

Set the time manually
`# clock set hh:mm:ss day_of_month 3_letter_month year`

Configure the IOS device to be a NTP server
`(config)# ntp master number`

Get time from a NTP server
`(config)# ntp server ip_address`

17 Syslog

Make logged events display the data and time associated with them
`(config)# service timestamps log datetime`

Configure the destination hostname or IP address of the syslog server
`(config)# logging ip_address`

Control the messages that will be sent to the syslog server. Fore example, to limit, the messages to levels 4 and lower (0 to 4), use one of the two equivalent commands:
`(config)# logging trap 4`
`(config)# logging trap warning`

Optional. This specifies that syslog packets contain the IPv4 or IPv6 address of a specific

[18] Specifies the IP address of the tunnel interface.

interface, regardless of which interface the packet uses to exit the router.

```
(config)# logging source-interface interface_id
```

17.1 Verify

```
show logging | include changed state to up
show logging | begin June 12 22:35
```

17.2 Syslog Severity Level

Severity Name	Severity Level	Explanation
Emergency	Level 0	System Unusable
Alert	Level 1	Immediate Action Needed
Critical	Level 2	Critical Condition
Error	Level 3	Error Condition
Warning	Level 4	Warning Condition
Notification	Level 5	Normal, but Significant Condition
Information	Level 6	Information Message
Debugging	Level 7	Debugging Message

18 ACL - Access Control Lists

18.1 Standard ACL

Matches based only on source address.

Place them as close the destination as possible.

The full syntax of the standard ACL command is as follows:

```
(config)# access-list access_list_number {deny | permit | remark}
source [source-wildcard] [log]
```

Parameter	Description
access_list_number	Number of an ACL. This is a decimal number from 1 to 99 or 1300 to 1999 (for standard ACL).
deny	Denies access if the condition are matched.
permit	Permits access id the conditions are matched.
remark	Add a remark about entries in an IP access list to make the list easier to understand and scan.
source	Number of the network or host from which the packet is being sent. There are two ways to specify the *source*: • Use a 32-bit quantity in four-part, dotted-decimal format. • Use the keyword **any** as an abbreviation for a *source* and *source_wildcard* of 0.0.0.0 255.255.255.255.
source_wildcard	(Optional) 32-bit wildcard mask to be applied to the source. Places ones in the bit positions you want to ignore.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the `logging console` command.)

After a standard ACL is configured, it is linked to an interface using the `ip access-group` command in interface configuration mode:

`(config-if)#` **`ip access-group`** {*access_list_number* | *access_list_name*} { **in** | **out** }

18.1.1 Named Standard ACL

Alphanumeric name string must be unique and cannot begin with a number.

`(config)#` **`ip access-list`** [**standard** | **extended**] *name*

`(config-std-nacl)#` [**permit** | **deny** | **remark**] {*source* [*source_wildcard*]} [**log**]

Activate the named IP ACL on an interface

`(config-if)#` **`ip access-group`** *name* [**in** | **out**]

18.2 Extended ACL

Matches based on source/destination address, protocol, source/destination port number. Apply them as close to source as possible.

`(config)#` **`access-list`** *access_list_number* {**deny** | **permit** | **remark**} *protocol* [*source source-wildcard*] [*operator port* [*port_number or name*]] {*destination destination_wildcard*} [*operator port* [*port_number or name*]]

Parameter	Description
access_list_number	Identifies the access list using a number in the range 100 to 199 (for an extended IP ACL) and 2000 to 2699 (expanded IP ACLs).
remark	Used to enter a remark or comment.
protocol	Name or number of an Internet protocol. Common keywords include `icmp`, `ip`, `tcp` or `udp`. To match any Internet protocol (including ICMP, TCP, and UDP) use the `ip` keyword.
destination	Number of the network or host to which the packet is being sent
destination_wildcard	Wildcard bits to be applied to the destination.
operator	(Optional) Compares source or destination ports. Possible operands include `lt` (less than), `gt` (greater than), `eq` (equal), `neq` (not equal), and `range` (inclusive range).
port	(Optional) The decimal number or name of a TCP or UDP port.

18.3 ACL for IPv6

```
(config)# ipv6 access-list access_list_name
(config-ipv6-acl)# deny | permit protocol {source_ipv6_prefix/
prefix_length | any | host source_ipv6_address} [operator [port_
number]] {destination_ipv6_prefix/prefix_legnth | any | host
destination_ipv6_address} [operator [port_number]]
```

After an IPv6 ACL is configured, it is linked to an interface using the `ipv6 traffic-filter` command:

```
(config-if)# ipv6 traffic-filter access_list_name {in | out}
```

18.4 Show ACL

```
show access-lists
show ip access-lists
```

19 DHCP

```
(config)# ip dhcp excluded-address low_address [high_address]
(config)# ip dhcp pool pool_name
```

Required tasks

Define the address pool.

```
(dhcp-config)# network network_address [mask | /prefix_lenght]
```

Define the default router or gateway.

```
(dhcp-config)# default-router address [address2...address8]
```

Optional tasks

Define a DNS server.

```
(dhcp-config)# dns-server address [address2...address]
```

Define the domain name.

```
(dhcp-config)# domain-name domain
```

Define the duration of the DHCP lease.

```
(dhcp-config)# lease {days [hours] [minutes] | infinite}
```

Define the NetBIOS WINS server.

```
(dhcp-config)# netbios-name-server
```

19.1 Other

DHCP Client

```
(config-if)# ip address dhcp
```

DHCP Relay

~~ip helper-address~~

```
(config)# ip dhcp relay enable
(config)# ip dhcp relays server dhcp_ip_address
```

19.2 Troubleshoot

```
show ip dhcp binding
show ip dhcp server statistics
show ip dhcp conflict
show interfaces interface
show running-config | section dhcp
debug ip dhcp server events
```

20 NAT - Network Address Translation

20.1 Static NAT

```
(config)# ip nat inside source static local_ip global_ip
(config)# interface interface_id
(config-if)# ip nat inside
(config-if)# interface interface_id
(config-if)# ip nat outside
```

Static NAT with TCP port specified

Use a static NAT statement to redirect TCP port 80 traffic from 64.102.139.2(inside global) to 10.10.10.10(inside local):

```
(config)# ip nat inside source static tcp 10.10.10.10 80
64.102.139.2 80
```

20.2 Dynamic NAT

Define a pool of global addresses to be used for translation.

(config)# **ip nat pool** *name start_ip end_ip* {**netmask** *netmask* | **prefix-length** *prefix_lenght*}

Configure a standard access list permitting the addresses that should be translated.

(config)# **access-list** *access_list_number* **permit** *source* [*source_wildcard*]

Establish dynamic source translation, specifying the access list and pool defined in prior steps.

(config)# **ip nat inside source list** *access_list_number* **pool** *name*

Identify the inside interface.

(config)# **interface** *interface_id*
(config-if)# **ip nat inside**

Identify the outside interface.

(config)# **interface** *interface_id*
(config-if)# **ip nat outside**

20.3 PAT - Port Address Translation

Configure PAT for a sinlge public IPv4 address

Define a standard access list permitting the addresses that should be translated.

(config)# **access-list** *access_list_number* **permit** *source* [*source_wildcard*]

Establish dynamic source translation, specifying the ACL, exit interface and overload options.

(config)# **ip nat inside source list** *access_list_number* **interface** *interface_id* **overload**

Identify the inside interface.

(config)# **interface** *interface_id*
(config-if)# **ip nat inside**

Identify the outside interface.

(config)# **interface** *interface_id*

```
(config-if)# ip nat outside
```

Configure PAT for a pool of public IPv4 addresses

Define a pool of global addresses to be used for overload translation.

```
(config)# ip nat pool name start_ip end_ip {netmask netmask |
prefix-length prefix_length}
```

Define a standard access list permitting the addresses that should be translated.

```
(config)# access-list access-list-number permit source
[source_wildcard]
```

Establish overload translation, specifying the access list and pool defined in prior steps.

```
(config)# ip nat inside source list access_list_number pool name
overload
```

Identify the inside interface.

```
(config)# interface interface_id
(config-if)# ip nat inside
```

Identify the outside interface.

```
(config)# interface interface_id
(config-if)# ip nat outside
```

Show

```
show ip nat translations
show ip nat statistics
```