Unit 4
IPv4 Services & Implementations

1. Reset Password of Router
2. Backup Configuration in Router
3. Managing ACLs
   - Standard and Extended
   - Named and Numbered
4. Types of NAT
   - Static, Dynamic, PAT
5. Configuration of Named & Numbered ACLs
6. Configuration of NAT
   - Static, Dynamic, PAT

Set Router Password in Cisco Packet Tracer

1. Access the CLI

- Click on the router.
- Go to the **CLI tab**.
- --- System Configuration Dialog ---
  - Would you like to enter the initial configuration dialog? [yes/no]: no
- Press Enter to get started.

🔐 Set Console Password

Router> enable

Router# configure terminal

Router(config)# line console 0

Router(config-line)# password yourConsolePassword

Router(config-line)# login

Router(config-line)# exit

🔐 **Set Enable Password (to access privileged EXEC mode)**

Router(config)# enable password yourEnablePassword

✅ Or use enable secret (more secure as it is encrypted):

Router(config)# enable secret yourSecurePassword

💾 **Save Configuration**

Router# write memory

Or: do wr

Or: Router# copy running-config startup-config

Steps to Reset Password Using the Config Tab

1. **Click on the Router**
   → It opens the router settings window.
2. **Go to the "Config" Tab** (next to CLI)
3. In the left panel, find the **Settings** section.
4. Click the **Erase** button next to **NVRAM**.

✍️ **Reconfigure Passwords**

Router> enable

```
Router# configure terminal
Router(config)# enable secret myNewPassword
Router(config)# line console 0
Router(config-line)# password myNewConsolePass
Router(config-line)# login
Router# write memory
```

Backup Configuration in Cisco Router (Packet Tracer)

🔄 Step 1: Access the CLI

- Click on the **router**
- Go to the **CLI** tab

💾 Step 2: Save Current Running Configuration

```
Router# copy running-config startup-config
```

✅ This command ensures your current config is saved even after a reboot.

📤 Step 3: Backup to TFTP Server

(Assuming a TFTP server is connected and configured in your network)

```
Router# copy running-config tftp:
```

- **Address or name of remote host**: Enter TFTP Server IP
- **Destination filename**: Enter a name (e.g., R1-Backup)

✅ You should see a message like:

Writing R1-Backup

[OK]

📀 **Step 4: Backup Startup Configuration to TFTP**

Router# copy startup-config tftp:

Use this if you want to back up your saved configuration.

📩 **Step 5: Restore from TFTP (when needed)**

Router# copy tftp: running-config

- **Address or name of remote host**: TFTP Server IP
- **Source filename**: The name of your backup file

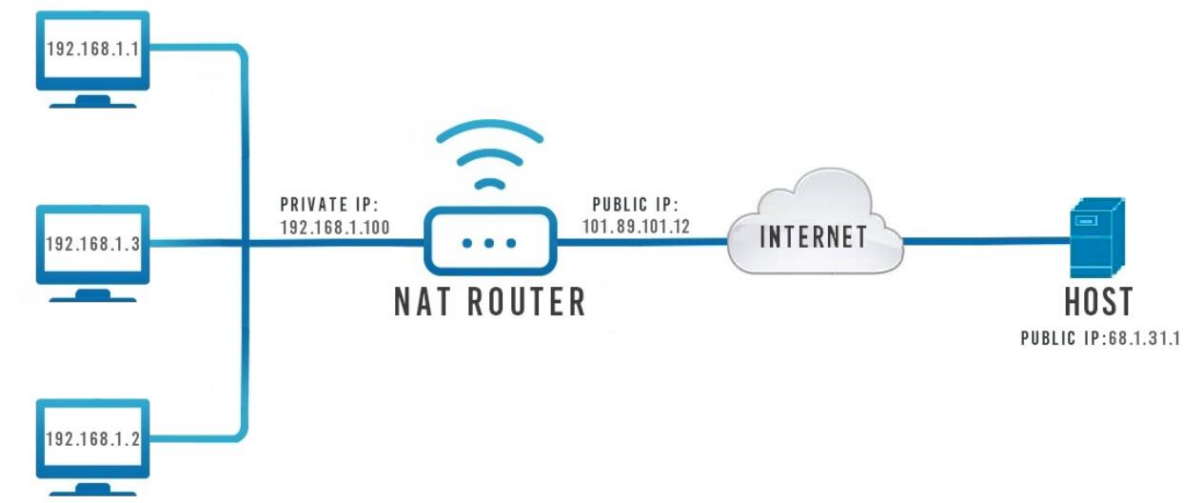
✅ Configuration will be applied to the router instantly.

What is NAT (Network Address Translation)?

https://www.practicalnetworking.net/series/nat/why-nat/

NAT is a technique that allows devices on a private network to communicate with the internet using a single public IP address, effectively hiding the internal network from the outside world and conserving (to avoid wasting something) IP addresses.

Network Address Translation is a method used in networking to **map private (local) IP addresses to public IP addresses** and vice versa. It is typically performed by a router or firewall.



How it works:

When a device on the private network wants to access the internet, the router (or firewall) intercepts the outgoing data, replaces the source IP address with its own public IP address, and forwards the packet. When the response comes back, the router intercepts it, replaces the destination IP address with the original private IP address of the device, and sends it to the correct device.

Why NAT is used?

- IPv4 addresses are limited, and NAT helps to **conserve public IP addresses**.
- Enhanced security: By hiding the internal network behind a single public IP address, NAT can help to protect the internal network from unauthorized access.
- Enables **multiple devices** on a LAN to access the internet using a **single public IP address**.
- Simplifies network configuration: NAT allows for the use of non-routable private IP addresses within a network, simplifying network configuration and management.

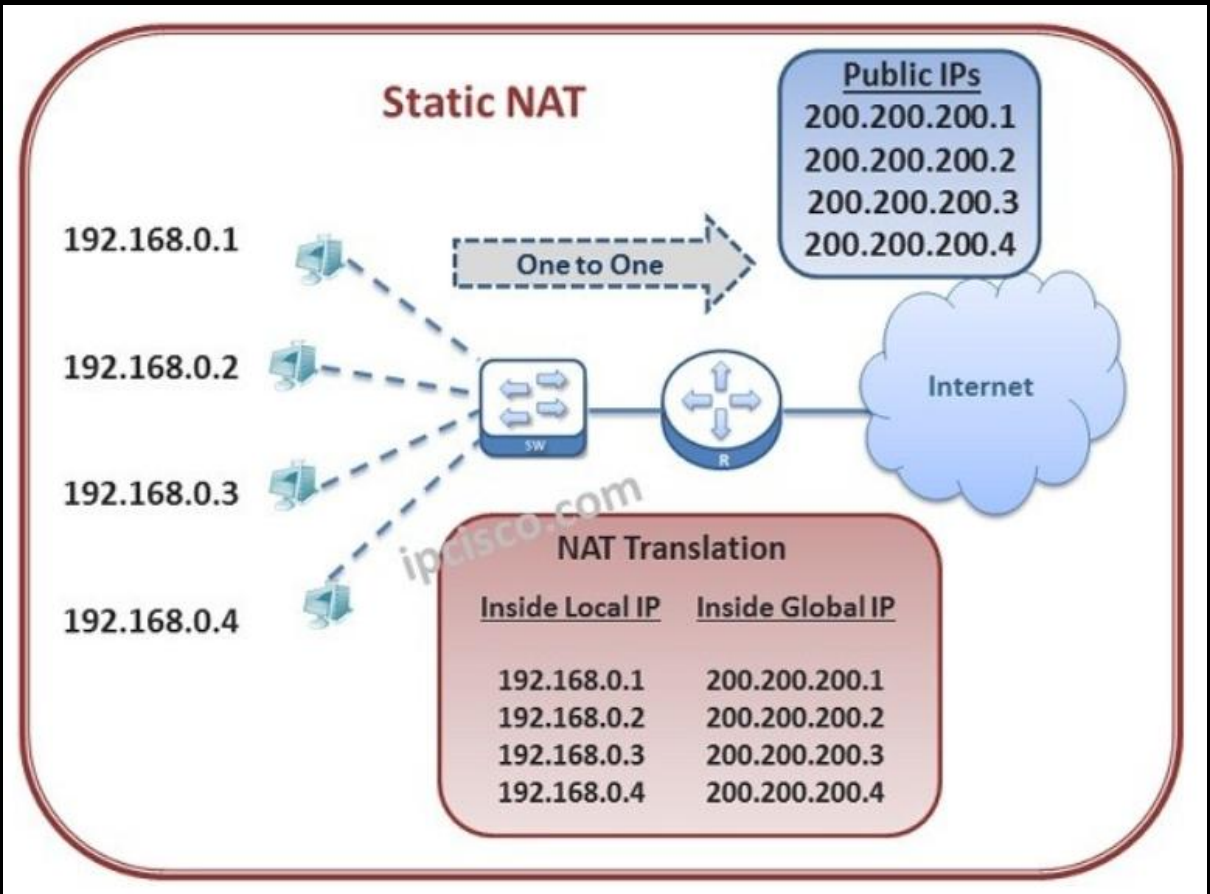
Types of NAT: (There are three main types of NAT)

1. Static NAT (One-to-One Mapping)

Maps a **single private IP address** to a **single public IP address**. It provides a one-to-one mapping of private IP addresses to public IP addresses, allowing internal network devices to access the internet while maintaining a predictable and persistent translation, useful for servers accessible from the internet.

📌 **Key Points:**

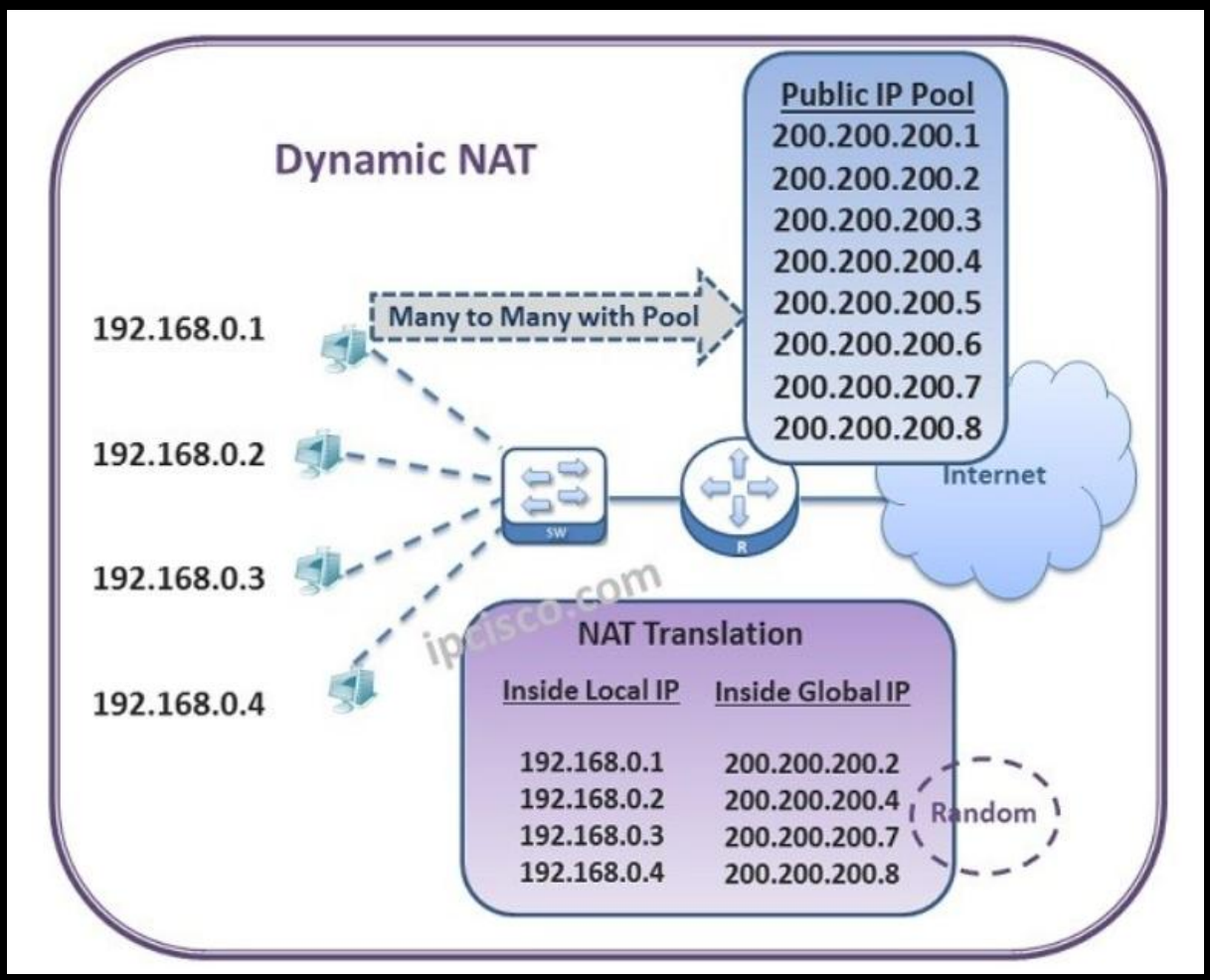
- The mapping is **fixed** and **manually configured**.
- Used when a device inside the network needs to be **consistently accessible from outside**, like a **web server** or **mail server**.



Advantages:	Disadvantages:
Simple and predictable.	Wastes public IPs.
Useful for hosting services.	Not scalable for large networks.

2. Dynamic NAT (Many-to-Many Mapping)

Maps **private IP addresses** to a **pool of public IP addresses**. These IP Addresses are given to the Internal users randomly. So, it is difficult to reach any Internal user from outside.



📌 **Key Points:**

- Mapping is **temporary** and done **dynamically**.
- **Dynamic NAT** is used when the number of internal Internet users are known.
- When an internal host accesses the internet, NAT **assigns an available public IP** from the pool.
- When the session ends, the public IP becomes **available again**.

Advantages:	**Disadvantages:**
Efficient use of public IP pool.	Cannot guarantee the same public IP every time.
Automatic mapping.	Still requires **multiple public IPs**.

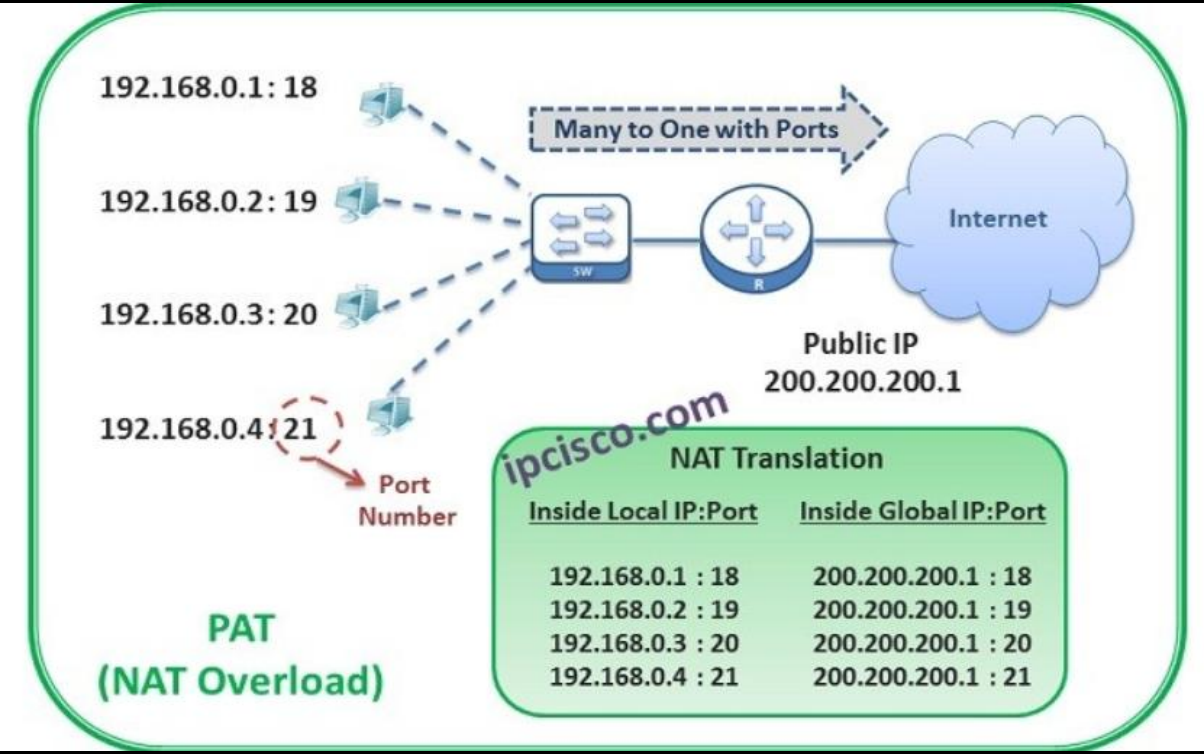
3. PAT (Port Address Translation) – Also called NAT Overload

Here, many Private IP Addresses are translated to one Public IP Address. The traffic distinguisher in PAT are Port Numbers, TCP/UDP ports are used in **PAT (NAT Overload).**

Each IP Address's traffic is determined by these ports. If you have many Intrenet user in a location, this type of NAT is very useful for you.

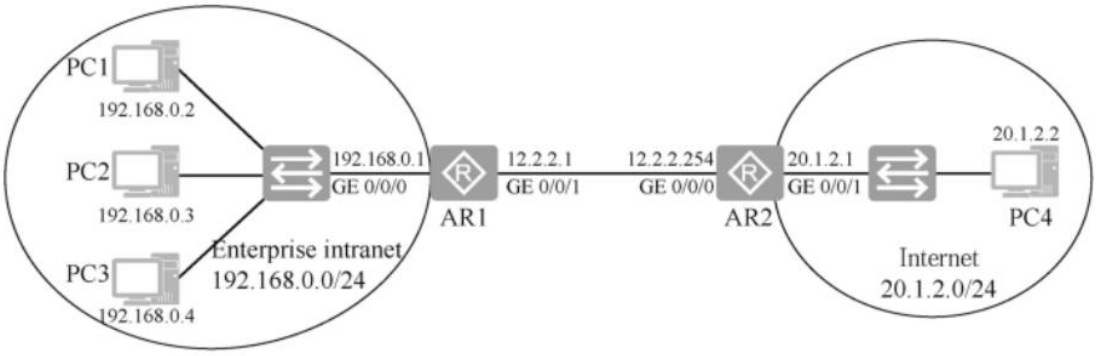
📌 **Key Points:**

- Most **common form of NAT** used in home/office routers.
- Uses **port numbers** to differentiate between sessions.
- Also known as **NAT Overload**.

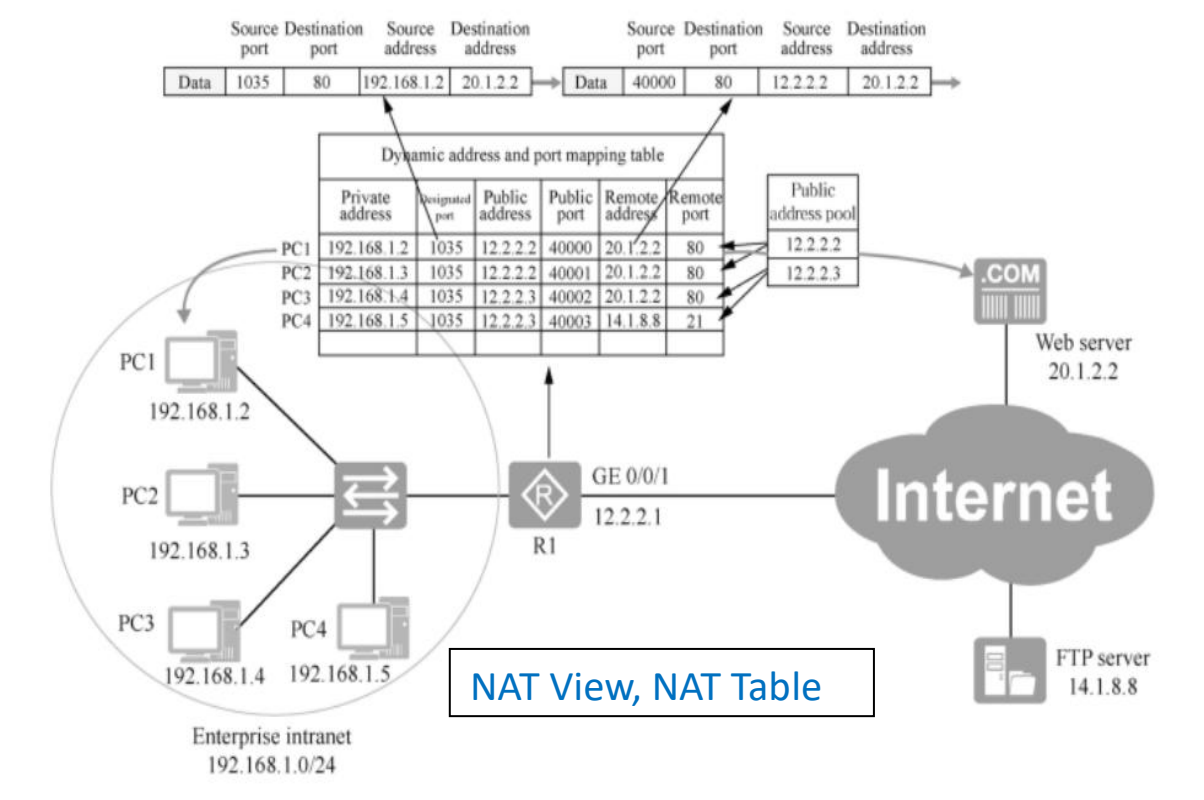


Advantages:	Disadvantages:
Very efficient.	Limited number of ports.
Only one public IP is needed for many users.	Some applications (VoIP, online gaming) may have issues.
Good for home/small office networks.	

https://link.springer.com/chapter/10.1007/978-981-19-3029-4_10



Configure static NAT



Summary Table

Type	Mapping	Public IP Usage	Port Translation	Use Case
Static NAT	One-to-One	High	No	Hosting servers (fixed IP)
Dynamic NAT	Many-to-Many	Medium	No	Office with public IP pool
PAT	Many-to-One	Low	Yes	Home/small network internet

🔐 Network Access Control List (ACL)

A set of rules that determine which users or systems have access to specific network resources, acting as a gatekeeper for network traffic and enhancing security by controlling who can access what

An **Access Control List (ACL)** is used to **filter network traffic** and **control access** to and from a network. ACLs are applied on routers or firewalls to **permit or deny** traffic based on various conditions.

🎯 Purpose of ACL:

- **Improve security** by restricting access to specific resources.
- Control the **flow of traffic** into and out of a network.
- Act as **traffic filters** for IP packets based on criteria like source, destination, ports, protocols, etc.

📑 How ACL Works:

ACLs work by examining each packet's **header** and comparing it to a list of rules. Each rule either:

- ✅ **Permits** the packet
- ❌ **Denies** the packet

Once a match is found, the action is taken, and **no further rules are checked** (first match wins).

🧠 Types of Access Control Lists (ACLs)

ACLs are used to control the flow of traffic into or out of a router interface by allowing or denying packets based on defined rules. There are two main types:

✅ 1. Numbered ACLs

✅ 2. Named ACLs

🔢 1. Numbered ACLs

➤ Description:

- Identified by a number (instead of a name).
- Quick to configure but less readable.
- Limited flexibility when editing rules.

🔹 Standard Numbered ACLs

- Range: 1–99 and 1300–1999
- Function: Filters only by source IP address
- Use Case: Simple traffic filtering where protocol or destination is not relevant

📝 Syntax:

access-list [1–99 | 1300–1999] [permit|deny] [source IP] [wildcard mask]

✅ Example:

R1(config)# access-list 10 permit 192.168.1.0 0.0.0.255

R1(config)# interface fa0/0

R1(config-if)# ip access-group 10 in

- *Allows traffic from 192.168.1.0/24 on the inbound interface.*

🔹 Extended Numbered ACLs

- Range: 100–199 and 2000–2699

- Function: Filters by source/destination IP, protocol, port number
- Use Case: More complex traffic filtering (e.g., HTTP, SSH, etc.)

📓 Syntax:

access-list [100–199 | 2000–2699] [permit|deny] [protocol] [source IP] [wildcard mask] [destination IP] [wildcard mask] [port]

✅ Example:

R1(config)# access-list 110 permit tcp any host 192.168.2.10 eq 80

R1(config)# interface fa0/1

R1(config-if)# ip access-group 110 out

- *Allows TCP traffic to host 192.168.2.10 on port 80 (HTTP).*

🏷️ 2. Named ACLs

➤ Description:

- Identified by a custom name instead of a number.
- Easier to read, manage, and modify.
- Support both standard and extended ACL types.

🔹 Standard Named ACL

- Filters traffic by source IP only.

📓 Syntax:

ip access-list standard [NAME]
[permit|deny] [source IP] [wildcard mask]

✅ Example:

R1(config)# ip access-list standard ALLOW_ADMIN

R1(config-std-nacl)# permit 192.168.1.0 0.0.0.255

R1(config)# interface fa0/0

R1(config-if)# ip access-group ALLOW_ADMIN in

🔹 Extended Named ACL

- Filters traffic by source/destination IP, protocol, ports.

📓 Syntax:

ip access-list extended [NAME]

[permit|deny] [protocol] [source IP] [wildcard mask] [destination IP] [wildcard mask] [port]

✅ Example:

R1(config)# ip access-list extended WEB_TRAFFIC

R1(config-ext-nacl)# permit tcp any host 192.168.2.10 eq 80

R1(config-ext-nacl)# deny ip any any

R1(config)# interface fa0/1

R1(config-if)# ip access-group WEB_TRAFFIC out

- *Allows HTTP traffic to 192.168.2.10 and denies all other traffic.*

⚙️ Key Differences Between Numbered and Named ACLs

Feature	Numbered ACL	Named ACL
Identification	By number	By name
Readability	Low	High
Edit Flexibility	Not flexible (must retype)	Easily editable
Types Supported	Standard & Extended	Standard & Extended

🛠️ Tips for Configuration

- Use **named ACLs** for easier maintenance and troubleshooting.
- Use **wildcard masks** carefully (0.0.0.255 = /24).
- Always include a deny ip any any at the end of extended ACLs if you want to drop all other traffic.
- Apply ACLs in the correct **direction** (in or out) and on the correct **interface**.

📍 Where ACLs Are Applied:

ACLs can be applied:

- **Inbound** – Before the traffic enters the interface
- **Outbound** – After the traffic is processed by the router and is about to leave the interface

📝 Syntax Example:

ip access-group 10 in

✅ Key ACL Commands in Cisco (Example)

Router(config)# access-list 10 permit 192.168.1.0 0.0.0.255

Router(config)# interface fa0/0

Router(config-if)# ip access-group 10 in

🔄 Wildcard Mask in ACLs

- A wildcard mask is used to **specify a range of IP addresses**.
- It's the **opposite** of a subnet mask.

📝 Example:

192.168.1.0 0.0.0.255 → covers 192.168.1.0 to 192.168.1.255

📋 Important Notes:

- ACLs are processed **top-down** (first match is used).
- Implicit **deny all** at the end of every ACL.
- ACLs do **not state a default permit**—you must explicitly allow what you need.

🚧 ACL Use Cases:

Use Case	ACL Type Used
Allow only internal users	Standard ACL
Block access to a specific site	Extended ACL
Restrict FTP or SSH traffic	Extended ACL
Improve network performance	Standard/Extended

✅ Summary Table:

Type	Filters By	ACL Number Range	Example Usage
Standard	Source IP only	1–99, 1300–1999	Block users by IP
Extended	Src/Dest IP, Protocol, Ports	100–199, 2000–2699	Block ports like FTP, SSH
Named	Custom-named ACLs	Custom	Easy management, readable