ACCESS CONTROLS Tutorial

Faculty of Information Technology, Hanoi University

Contents

- What is an Access Control List?
- Why Use An ACL?
- Where Can You Place An ACL?
- What Are The Components of An ACL?
- What Are The Types of ACLs?
- How to Implement An ACL on a Router?

What is an Access Control List?

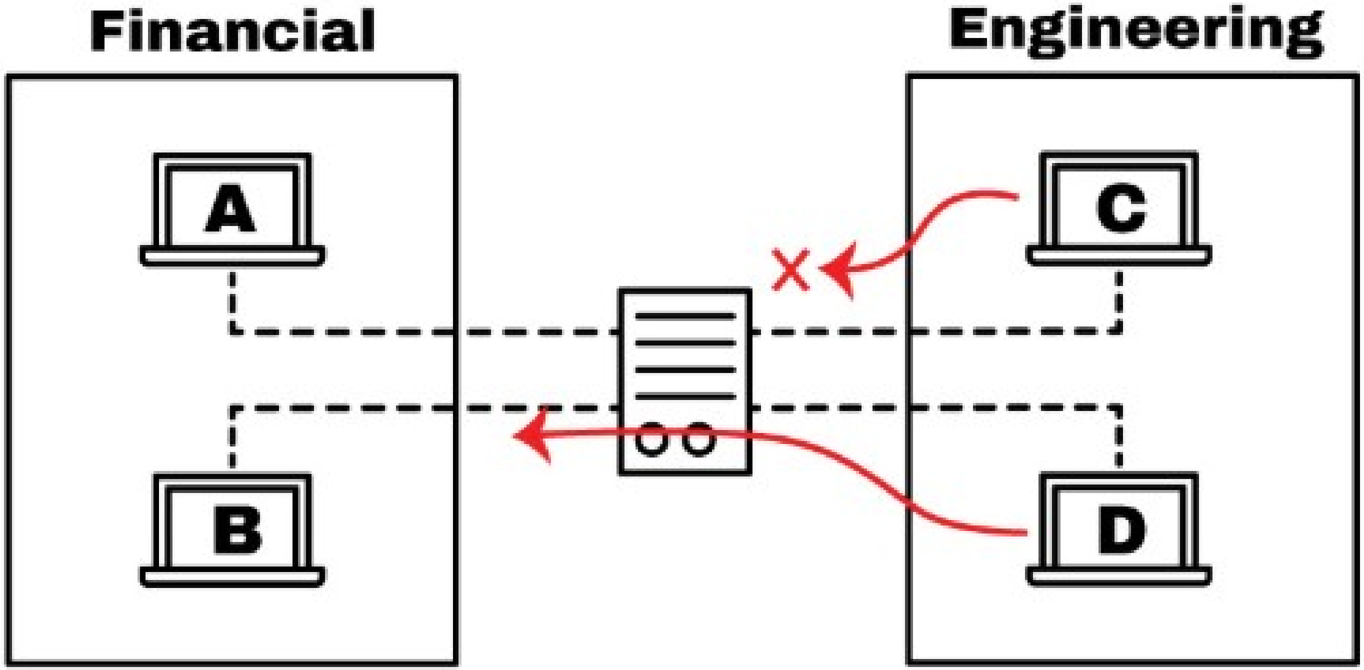
- In the computer networking world, an **ACL** is one of the most fundamental components of security.
- Access Control Lists “ACLs” are network traffic filters that can control incoming or outgoing traffic.
- ACLs work on a set of rules that define how to forward or block a packet at the router’s interface.
- An ACL is the same as a Stateless Firewall, which only restricts, blocks, or allows the packets that are flowing from source to destination.

Why Use An ACL?

- The main idea of using an ACL is to provide security to your network. Without it, any traffic is either allowed to enter or exit, making it more vulnerable to unwanted and dangerous traffic.

Why Use An ACL?

As shown in the picture below, the routing device has an ACL that is denying access to host C into the Financial network, and at the same time, it is all



Why Use An ACL?

- With an ACL you can filter packets for a single or group of IP address or different protocols, such as TCP or UDP.
- So for example, instead of blocking only one host in the engineering team, you can deny access to the entire network and only allow one. Or you can also restrict the access to host C.
- If the Engineer from host C, needs to access a web server located in the Financial network, you can only allow port 80, and block everything else

Where Can You Place An ACL?

The devices that are facing unknown external networks, such as the Internet, need to have a way to filter traffic. So, one of the best places to configure an ACL is on the edge routers.

What Are The Components of An ACL?

ACL is a set of rules or entries. You can have an ACL with single or multiple entries, where each one is supposed to do something, it can be to permit everything or block nothing.

What Are The Components of An ACL?

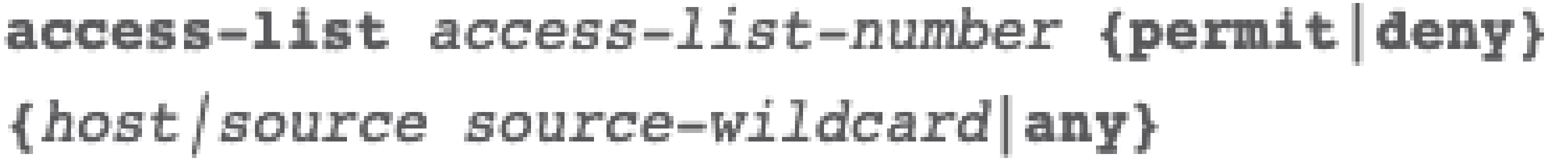
- **Sequence Number:**
  Identify an ACL entry using a number.
- **ACL Name:**
  Define an ACL entry using a name. Instead of using a sequence of numbers, some routers allow a combination of letters and numbers.
- **Remark:**
  Some Routers allow you to add comments into an ACL, which can help you to add detailed descriptions.
- **Statement:**
  Deny or permit a specific source based on address and wildcard mask. Some routing devices, such as Cisco, configure an implicit deny statement at the end of each ACL by default.

What Are The Components of An ACL?

- **Network Protocol:**
  Specify whether deny/permit IP, IPX, ICMP, TCP, UDP, NetBIOS, and more.
- **Source or Destination:**
  Define the Source or Destination target as a Single IP, a Address Range (CIDR), or all Addresses.
- **Log:**
  Some devices are capable of keeping logs when ACL matches are found.
- **Other Criteria:**
  Advanced ACLs allow you to use control traffic through the Type of Service (ToS), IP precedence, and differentiated services codepoint (DSCP) priority.

What Are The Types of ACLs?

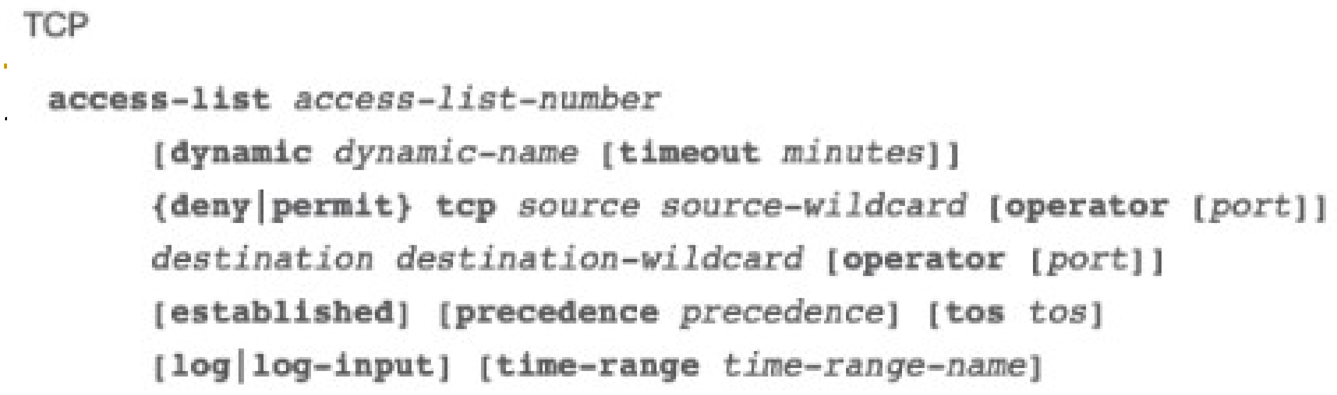
- The standard ACL aims to protect a network using only the source address.
- It is the most basic type and can be used for simple deployments, but unfortunately, it does not provide strong security. The configuration for a standard ACL on a Cisco router is as follows: (access-list-number: 1-99)



```
access-list access-list-number {permit|deny}
{host|source source-wildcard|any}
```

What Are The Types of ACLs?

- With the extended ACL, you can also block source and destination for single hosts or entire networks.
- You can also use an extended ACL to filter traffic based on protocol information (IP, ICMP, TCP, UDP).
- The configuration of an



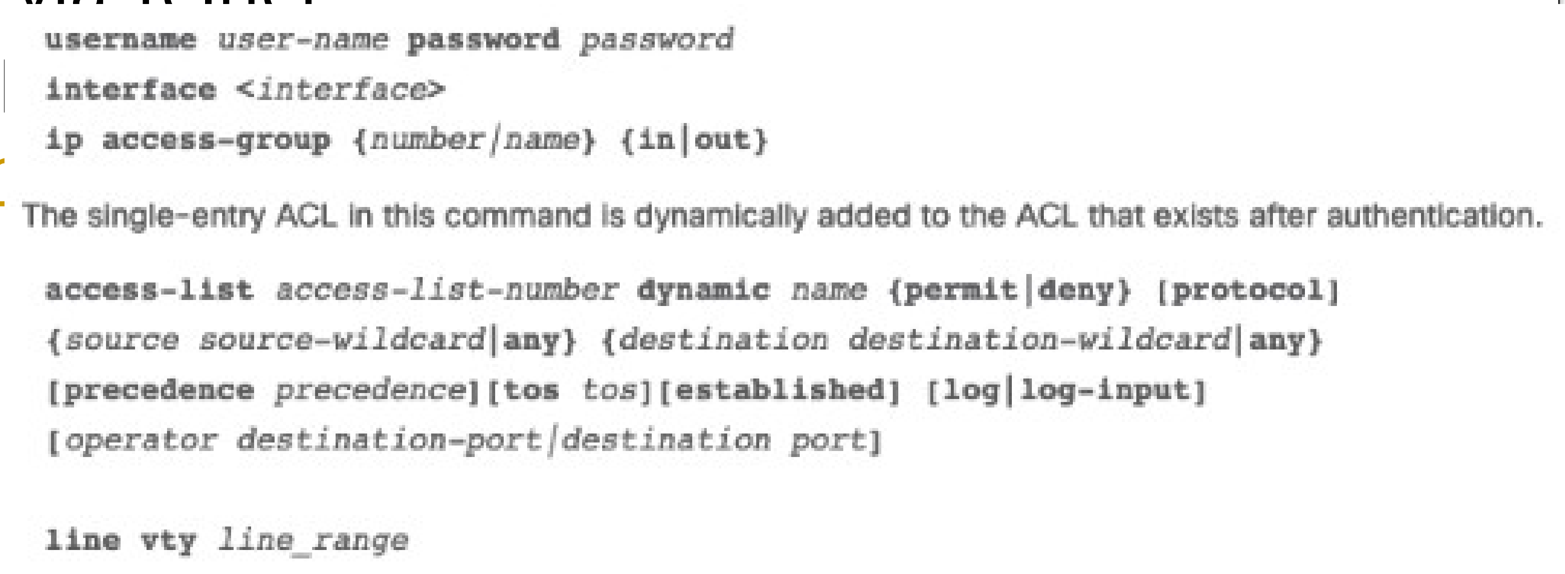
TCP

```
access-list access-list-number
        [dynamic dynamic-name [timeout minutes]]
        {deny|permit} tcp source source-wildcard [operator [port]]
        destination destination-wildcard [operator [port]]
        [established] [precedence precedence] [tos tos]
        [log|log-input] [time-range time-range-name]
```

as

What Are The Types of ACLs?

- Dynamic ACLs, rely upon extended ACLs, Telnet, and authentication. This type of ACLs are often referred to as “Lock and Key” and can be used for specific timeframes.
- These lists permit access to a user to a source or destination only if the user authenticates to the device via Telnet
- The fo
  Dynam



```
username user-name password password
interface <interface>
ip access-group {number/name} {in|out}
```

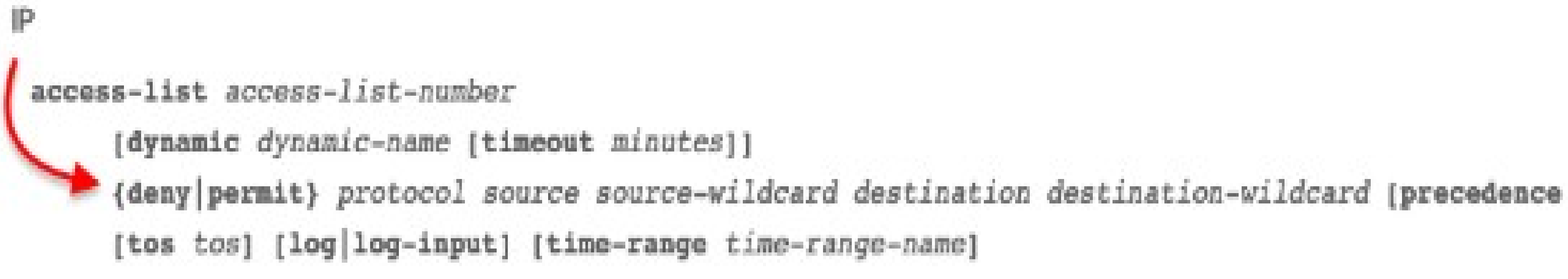
The single-entry ACL in this command is dynamically added to the ACL that exists after authentication.

```
access-list access-list-number dynamic name {permit|deny} [protocol]
{source source-wildcard|any} {destination destination-wildcard|any}
[precedence precedence][tos tos][established] [log|log-input]
[operator destination-port|destination port]

line vty line_range
```

How to Implement An ACL On your RouterS?

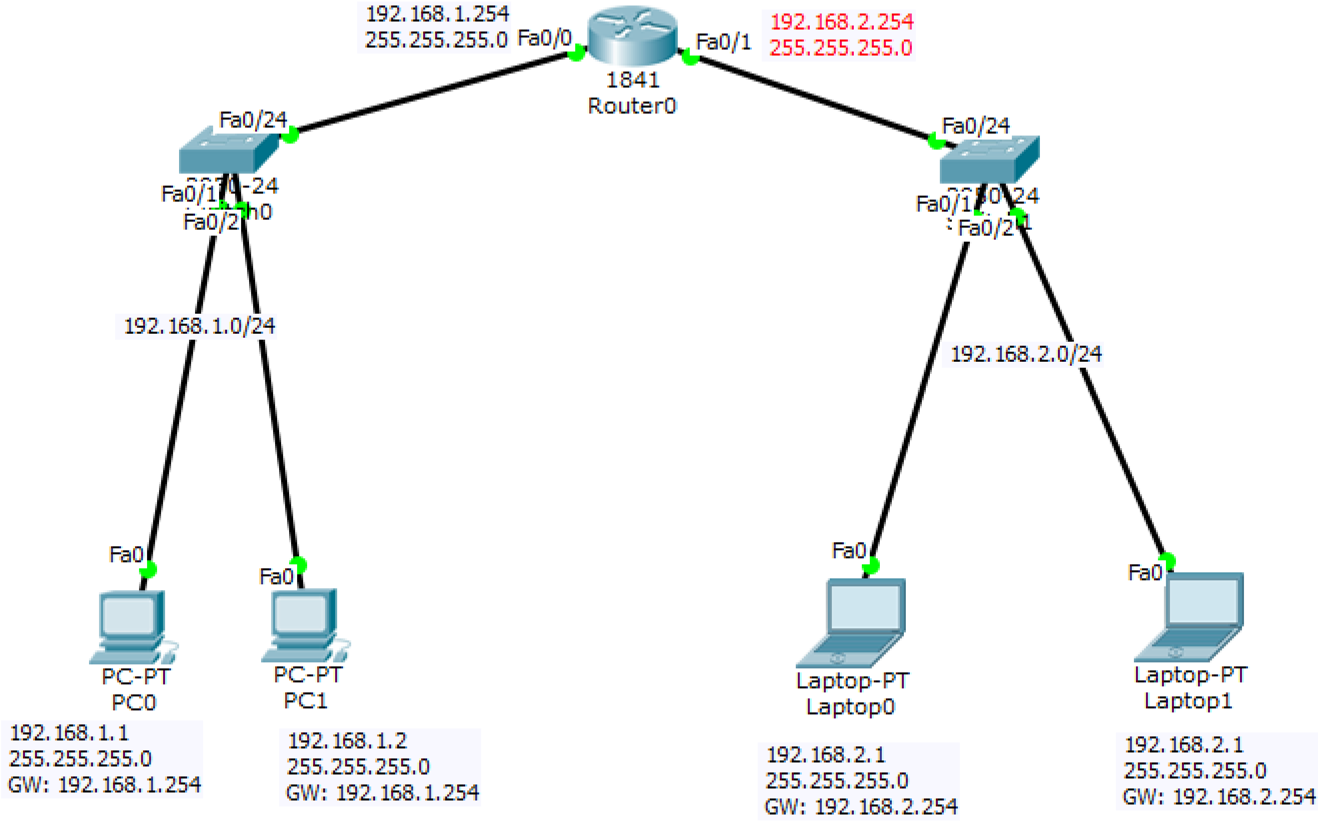
- For an ACL to work, apply it to a router’s interface. Since all routing and forwarding decisions are made from the router’s hardware, the ACL statements can be executed much faster
- When you create an ACL entry, the source address goes first, and the destination goes after. Take the example of the extended ACL configuration for IP on a Cisco Router. When you create a Deny/Permit rule, you must first define the source, and then the destination IP.



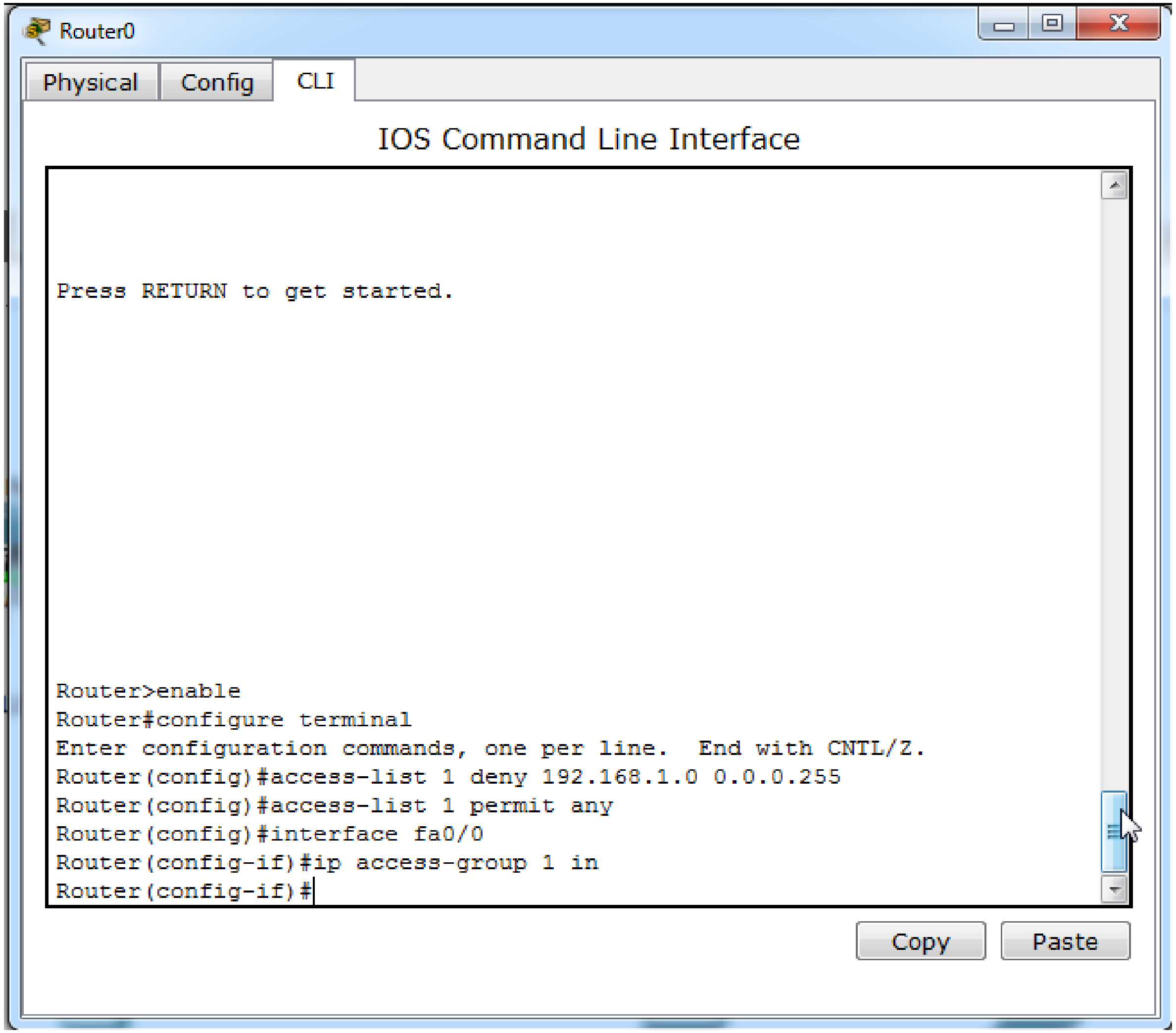
IP

```
access-list access-list-number
     [dynamic dynamic-name [timeout minutes]]
     {deny|permit} protocol source source-wildcard destination destination-wildcard [precedence
     [tos tos] [log|log-input] [time-range time-range-name]
```

Create 2 LANs connected together via 1 router



ACL at router



```
Press RETURN to get started.









Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#access-list 1 deny 192.168.1.0 0.0.0.255
Router(config)#access-list 1 permit any
Router(config)#interface fa0/0
Router(config-if)#ip access-group 1 in
Router(config-if)#
```