# Access Control Lists

*CyberQuince*

# Access Control Lists

An ***Access Control List*** (ACL) is a list of rules configured on a network device, used to filter traffic flowing through that device.

**Each packet** that goes through that device is compared against **each rule** in the ACL.

**If the packet doesn't match** the ***current*** rule, it's compared to the ***next*** one.

As soon as the packet matches a rule, the rule is applied and all the rules that come after are ignored.

**At the end of the list**, there usually is a "***catch all***" rule, that applies to packets that don't match any of the other rules.

# Configuring ACLs on Cisco Devices

**Any number of ACLs** can be created on a router (or other network device), but a list will not filter traffic **until it's applied to an interface**.

It's also not enough to only apply an ACL to a chosen interface, the ***direction*** of packet flow must also be specified.

# Configuring ACLs on Cisco Devices

**Any number of ACLs** can be created on a router (or other network device), but a list will not filter traffic **until it's applied to an interface**.

It's also not enough to only apply an ACL to a chosen interface, the ***direction*** of packet flow must also be specified.

***Inbound ACL*** = filters incoming traffic (packets are dropped/forwarded prior to being routed)

***Outbound ACL*** = filters outgoing traffic (packets are dropped/forwarded after being routed)

# Standard vs Extended ACLs

**Standard** ACLs can filter traffic based only on source IP.

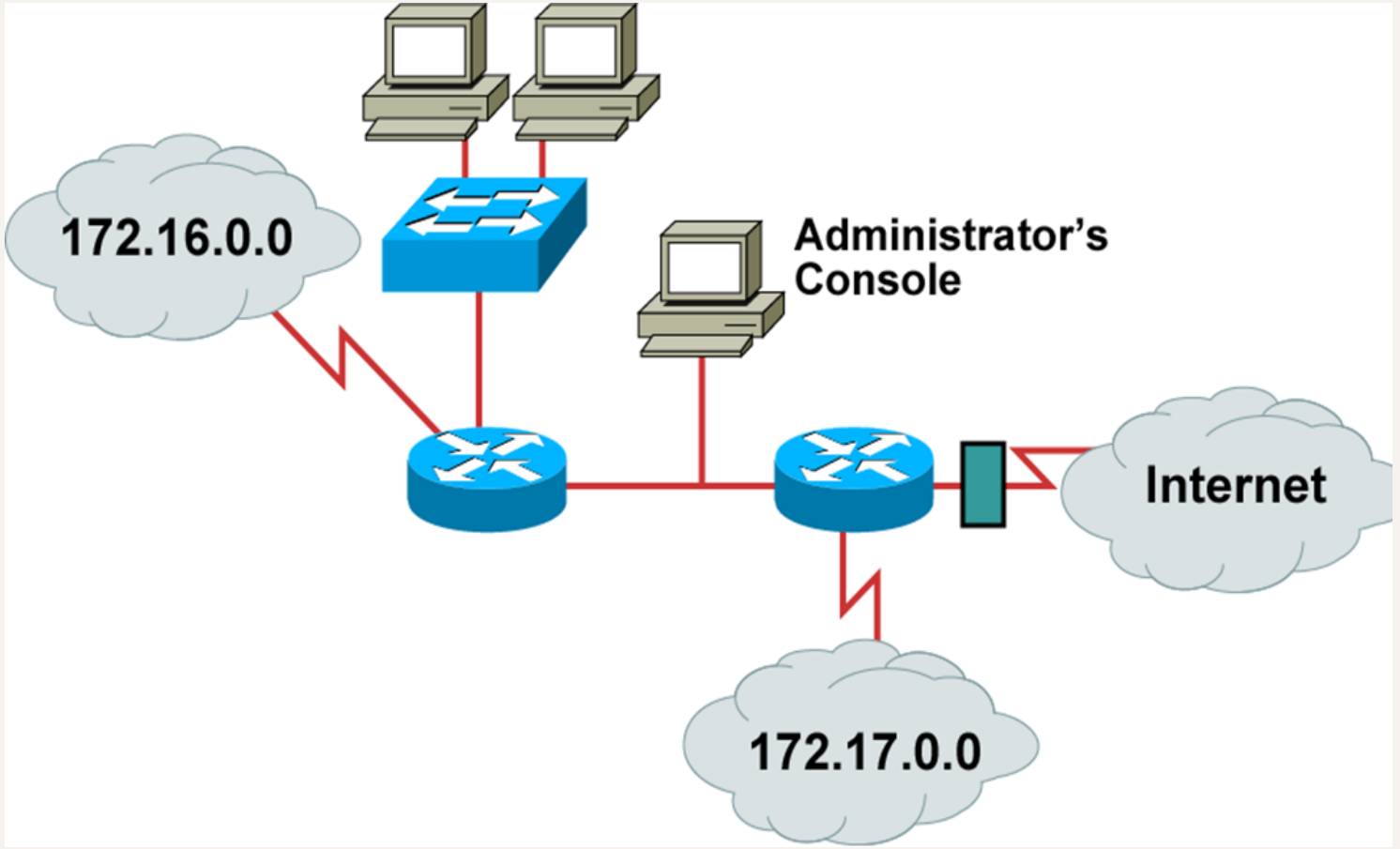
**Extended** ACLs can filter traffic based on both source and destination IPs, but also ports and protocols.

***For standard ACLs***, we use numbers 1-99, and 1000-1399 were added additionally;

***For extended ACLs***, we use numbers 100-199, and 2000-2699 were added additionally.

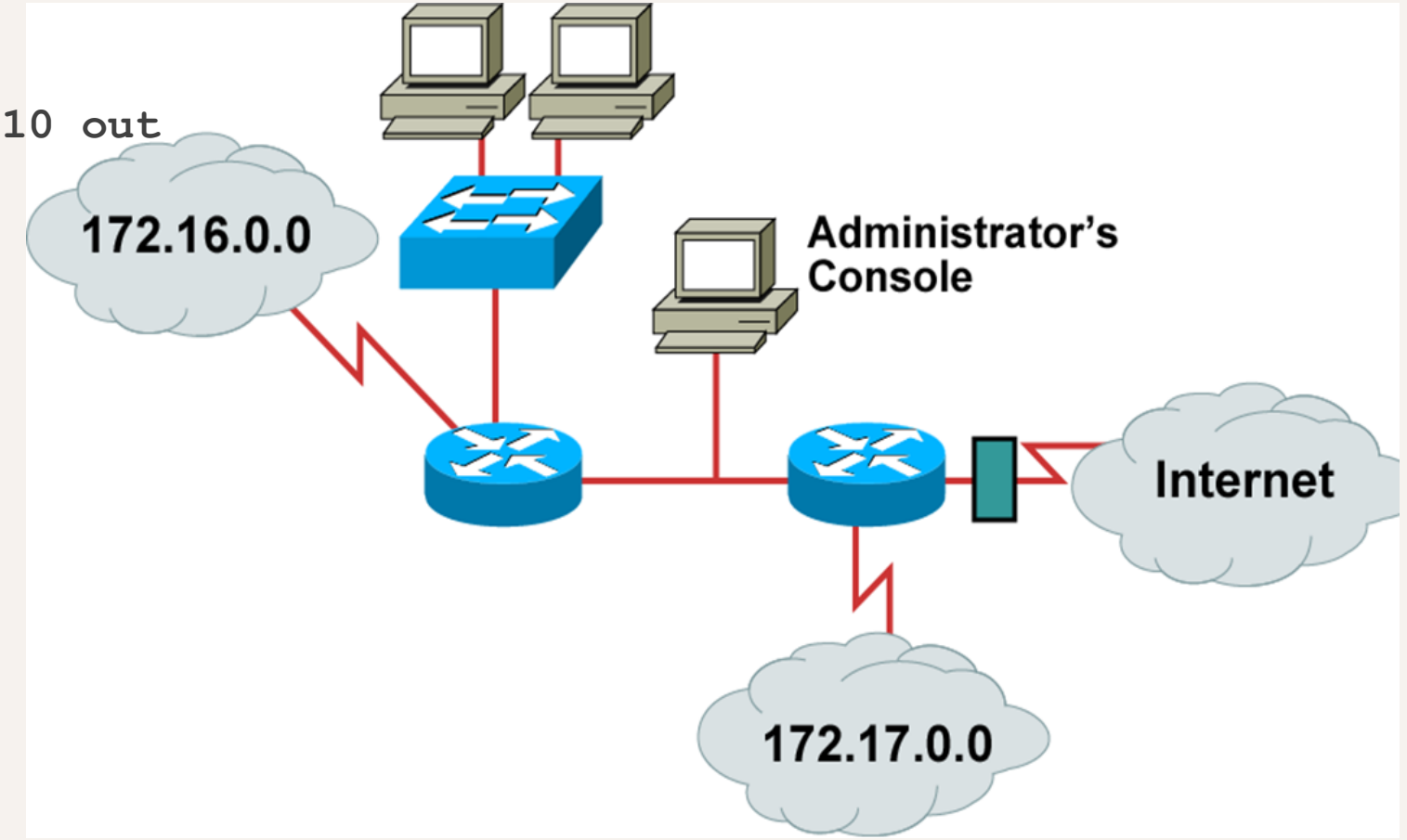
# Standard ACLs - Example

Computers from network 172.16.0.0 should not be able to reach Internet.



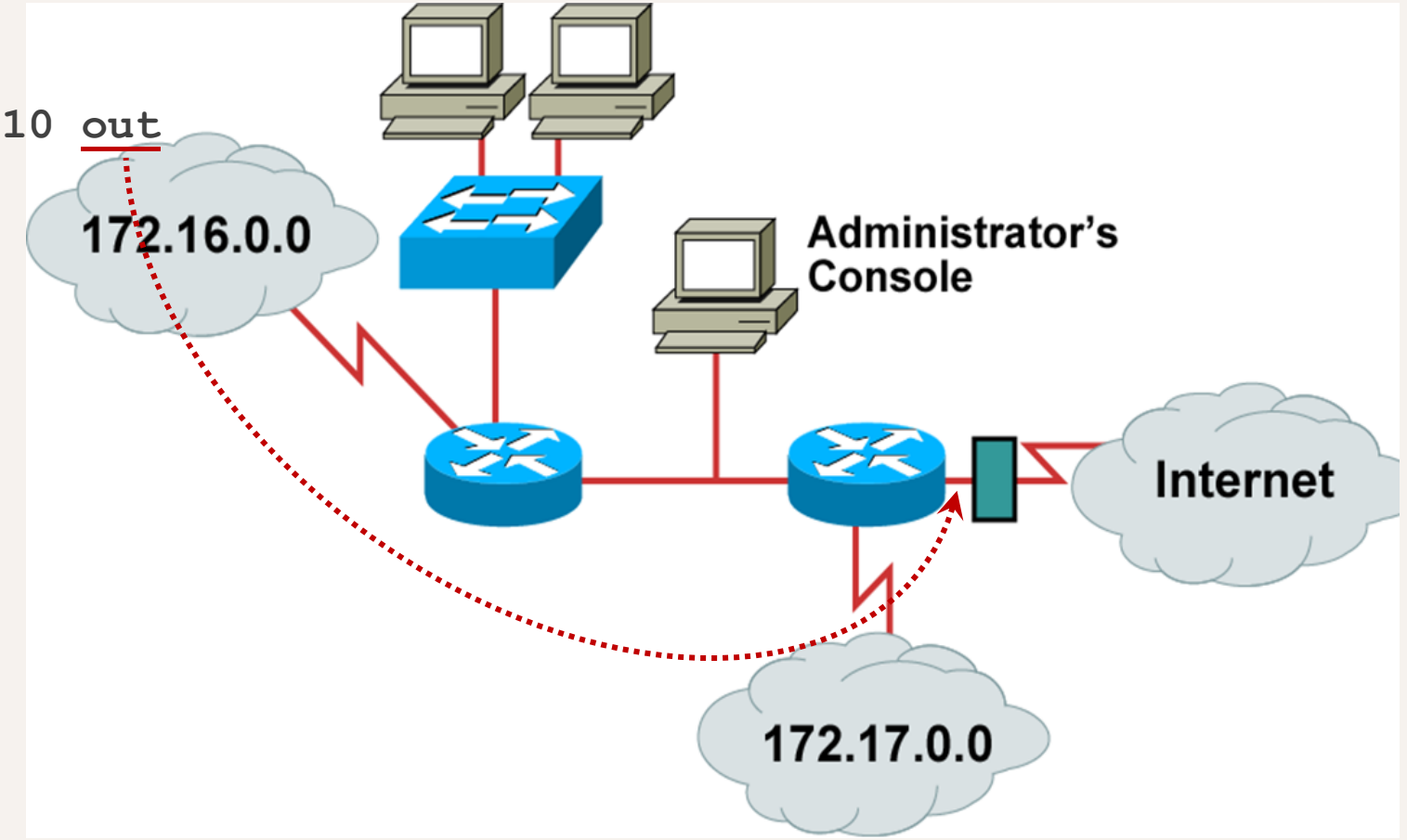
# Standard ACLs - Example

```
R(config)#access-list 10 deny 172.16.0.0 0.0.0.255
R(config)#access-list 10 permit any

R(config-if)#ip access-group 10 out
```



# Standard ACLs - Example

```
R(config)#access-list 10 deny 172.16.0.0 0.0.0.255
R(config)#access-list 10 permit any

R(config-if)#ip access-group 10 out
```



# ACLs — Different Ways of Configuration

```
R(config)#access-list 10 deny 172.16.0.0 0.0.0.255
R(config)#access-list 10 permit any
```

# ACLs — Different Ways of Configuration

```
R(config)#access-list 10 deny 172.16.0.0 0.0.0.255
R(config)#access-list 10 permit any

R(config)#ip access-list standard 10
R(config-std-nacl)#deny 172.16.0.0 0.0.0.255
R(config-std-nacl)#permit any
```

# ACLs — Different Ways of Configuration

```
R(config)#access-list 10 deny 172.16.0.0 0.0.0.255
R(config)#access-list 10 permit any

R(config)#ip access-list standard 10
R(config-std-nacl)#deny 172.16.0.0 0.0.0.255
R(config-std-nacl)#permit any

R(config)#ip access-list standard QuincesList
```

# ACLs — Different Ways of Configuration

*Permit or deny an entire subnet:*

```
R(config-std-nacl)#permit 192.168.0.0 0.0.0.255
R(config-std-nacl)#deny 192.168.0.0 0.0.0.255
```

# ACLs — Different Ways of Configuration

*Permit or deny an entire subnet:*

```
R(config-std-nacl)#permit 192.168.0.0 0.0.0.255
R(config-std-nacl)#deny 192.168.0.0 0.0.0.255
```

*Permit or deny a single host IP:*

```
R(config-std-nacl)#deny 192.168.0.15 0.0.0.0
R(config-std-nacl)#permit host 192.168.0.15
```

# ACLs — Different Ways of Configuration

*Permit or deny an entire subnet:*

```
R(config-std-nacl)#permit 192.168.0.0 0.0.0.255
R(config-std-nacl)#deny 192.168.0.0 0.0.0.255
```

*Permit or deny a single host IP:*

```
R(config-std-nacl)#deny 192.168.0.15 0.0.0.0
R(config-std-nacl)#permit host 192.168.0.15
```

*Permit or deny all:*

```
R(config-std-nacl)#deny 0.0.0.0 255.255.255.255
R(config-std-nacl)#permit any
```

# ACL — Some Rules…

1. There can be **only one** Access List ***per interface, per direction***

# ACL — Some Rules…

1. There can be **only one** Access List ***per interface, per direction***
2. Whenever a rule is added, it's added ***to the "bottom"*** of the list

# ACL — Some Rules…

1. There can be **only one** Access List ***per interface, per direction***
2. Whenever a rule is added, it's added ***to the "bottom"*** of the list
3. Every list has an "deny any" rule at the end

# ACL — Some Rules…

1. There can be **only one** Access List ***per interface, per direction***
2. Whenever a rule is added, it's added ***to the "bottom"*** of the list
3. Every list has an "deny any" rule at the end
4. If ***no "permit" rules*** are added to the list, it will block ***all traffic***

# ACL — Some Rules…

1. There can be **only one** Access List ***per interface, per direction***
2. Whenever a rule is added, it's added ***to the "bottom"*** of the list
3. Every list has an "deny any" rule at the end
4. If ***no "permit" rules*** are added to the list, it will ***block all traffic***
5. An empty list, applied to an interface, will not block traffic

# ACL — Some Rules…

1. There can be **only one** Access List ***per interface, per direction***
2. Whenever a rule is added, it's added ***to the "bottom"*** of the list
3. Every list has an "deny any" rule at the end
4. If ***no "permit" rules*** are added to the list, it will ***block all traffic***
5. An empty list, applied to an interface, will not block traffic
6. Traffic ***generated*** by a given router, ***will not be blocked*** by an ACL on that router

# ACL — Rule #6 Example


```
access-list 1 deny 10.0.0.0
0.255.255.255
!
interface FastEthernet0/0
 ip address 10.0.0.1 255.0.0.0
 ip access-group 1 in
 ip access-group 1 out
!
```

**1.** R1 pings R2 (whose IP is 10.0.0.2)

**2.** R1 sends an ICMP to R2, meaning, egress ACL will not drop the packet despite it matches the „deny 10.0.0.0/8“ rule

**3.** R2 will send the ping back to R1

**4.** Ingress ACL on interface F 0/0 will filter the packet and drop it

# Access Control Lists

*CyberQuince*