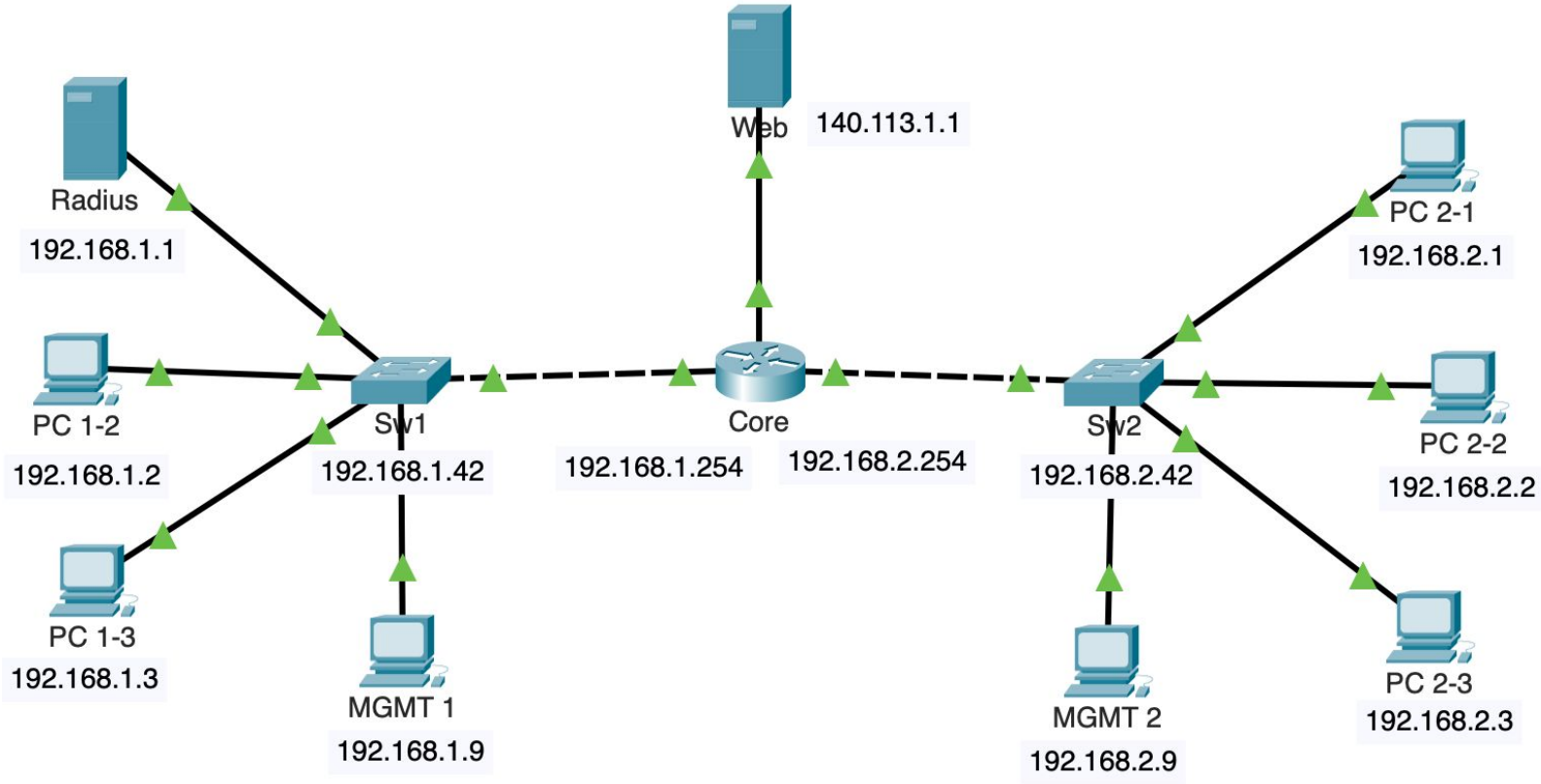
Lab 13. AAA, ACL

phkoan 管培勛
Credit to weiyuh, tsengch

Purpose

- Configure remote access on Switch & Router
- How to use Access Control List (ACL)

Topology


Web
140.113.1.1
Radius
192.168.1.1
PC 2-1
192.168.2.1
PC 1-2
192.168.1.2
Sw1
192.168.1.42
Core
192.168.1.254
192.168.2.254
Sw2
192.168.2.42
PC 2-2
192.168.2.2
PC 1-3
192.168.1.3
MGMT 1
192.168.1.9
MGMT 2
192.168.2.9
PC 2-3
192.168.2.3

AAA
(Authentication, Authorization and Accounting)

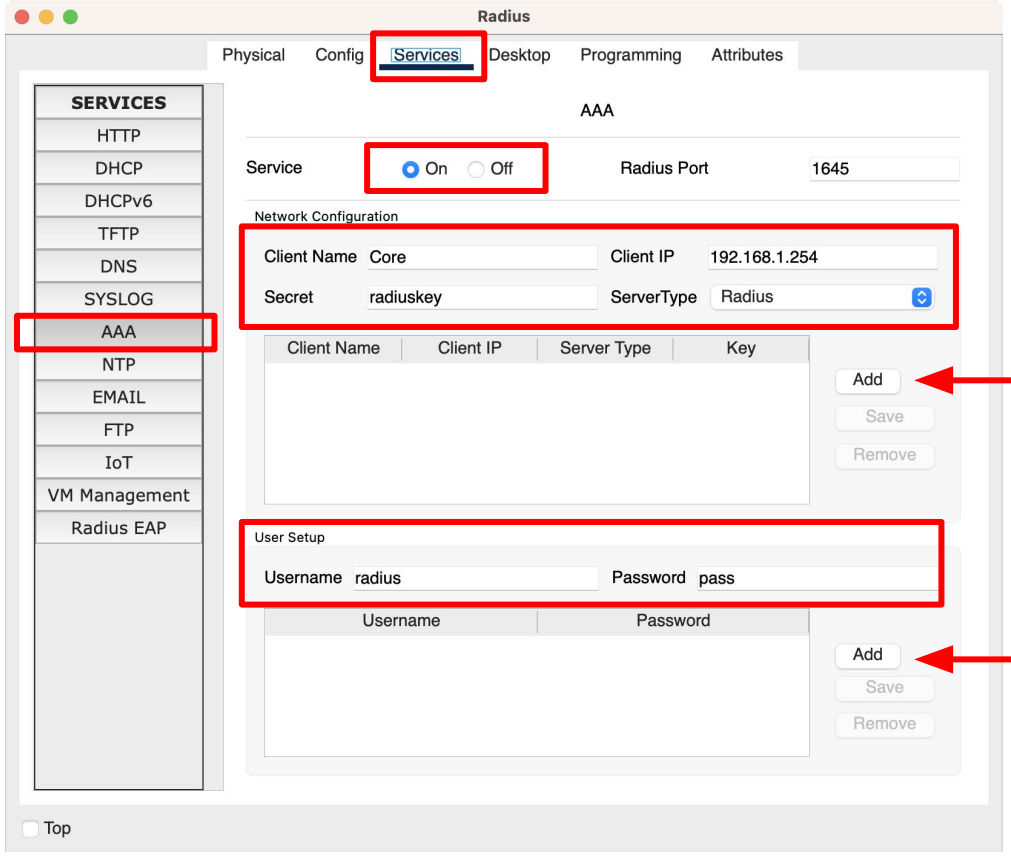
AAA (Authentication, Authorization and Accounting)

- **Auth**e**n**tication
  - Identify a user
- **Auth**ori**z**ation
  - Determine whether the user has the privilege to do something
- Accounting
  - Record who do what at when
- AAA can be configured at
  - Local
  - Remote server

RADIUS

- **R**emote **A**uthentication **D**ial-**I**n **U**ser **S**ervice
- A network protocol providing AAA management

RADIUS Server Configuration



RADIUS for Router - Configuration (1/2)

- Enable AAA

```
(config)# aaa new-model
```

- Enter radius server config mode

```
(config)# radius server server-name
```

- Configure server IP address

```
(config-radius-server)# address ipv4 server-ip [auth-port server-port]
```

- Add radius key

```
(config-radius-server)# key secret
```

RADIUS for Router - Configuration (2/2)

- Configure authentication method list

```
(config)# aaa authentication login list-name auth-list
```

- Apply the list to connections (e.g. console, vty, ...)

```
(config-line)# login authentication list-name
```

RADIUS for Router - Example

```
Core(config)# aaa new-model
Core(config)# radius server radius
Core(config-radius-server)# address ipv4 192.168.1.1
Core(config-radius-server)# key radiuskey
```

- Configure authentication provider group

```
Core(config)# aaa authentication login ccna group radius local
Core(config)# line vty 0 15
Core(config-line)# login authentication ccna
```

RADIUS for 2960 - Configuration

- Enable AAA

```
(config)# aaa new-model
```

- Configure radius server

```
(config)# radius-server host server-ip key secret
```

- Configure authentication method list

```
(config)# aaa authentication login list-name auth-list
```

- Apply the list to connections (e.g. console, vty, ...)

```
(config-line)# login authentication list-name
```

RADIUS for 2960 - Example

```
Core(config)# aaa new-model
Core(config)# radius-server host 192.168.1.1 key radiuskey
```

- Configure authentication method list

```
Core(config)# aaa authentication login ccna group radius local
Core(config)# line vty 0 15
Core(config-line)# login authentication ccna
```

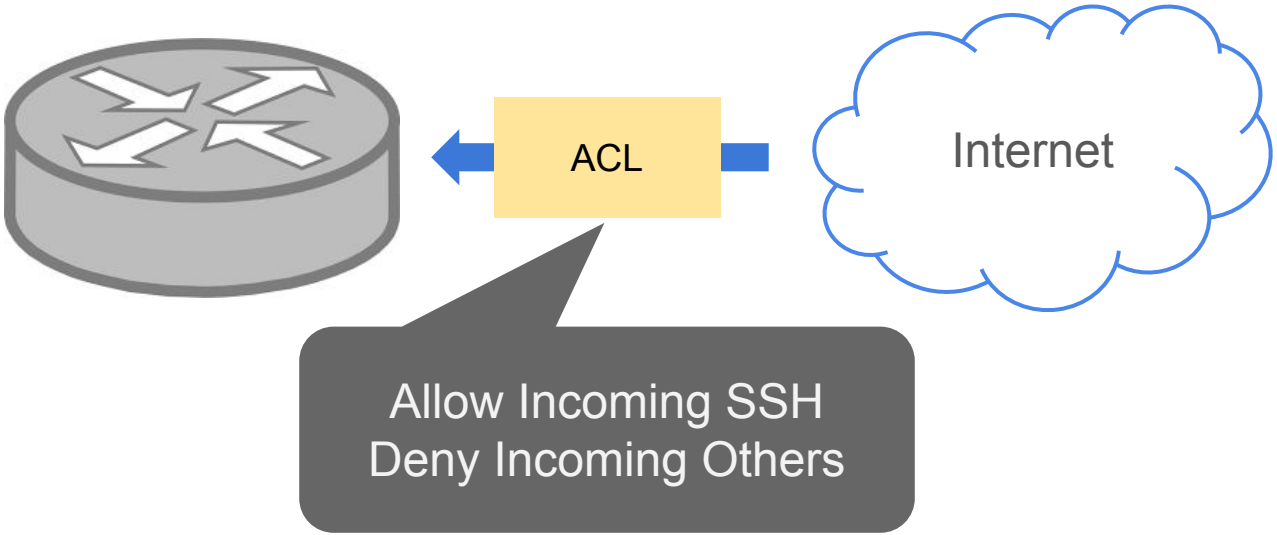
ACL
Access Control List

Access Control List (ACL)

- A sequential list of permit or deny statements, known as access control entries (ACEs)
- Control whether a switch/router permits or denies packets based on information found in the packet header

```
ip access-list standard access-2960
     10 deny host 140.113.0.229
     20 permit    192.168.0.0    0.0.0.255
     30 deny      192.168.235.0  0.0.0.63
     40 permit    192.168.235.0  0.0.0.255
     deny       any     # implicit deny
```

Access Control List (ACL) - Example



ACL - How To Configure

1. Create an ACL containing some ACEs
2. Apply the ACL to interface or line

Access Control List (ACL) - How it works

- ACL being applied to an interface
  - Router evaluates all network packets passing through the interface based on the ACL
  - Compare in sequential order, stop when matched (first match)
  - Either deny or permit
- Source IPv4 address is the main filtering criteria
- The last statement of an ACL is always an implicit deny
  - Block all traffic if no entries being matched

Access Control List (ACL) - Wildcard Masking

- 32-bit string
  - To determine which bits of the address to examine for a match
- Wildcard masks use the following rules to match binary 1s and 0s:
  - Wildcard mask bit `0` - Match the corresponding bit value in the address
  - Wildcard mask bit `1` - Ignore the corresponding bit value in the address
- Wildcard masks are often referred to inverse mask
  - Usually considered as opposite of subnet mask
  - But can be non-contiguous
- Wildcard mask keywords
  - **host:** `0.0.0.0`
  - **any:** `255.255.255.255`

Access Control List (ACL) - Wildcard Masking

	Decimal	Binary
IP address	192.168.1.1	11000000.10101000.00000001.00000001
Subnet Mask	**255.255.255.0**	11111111.11111111.11111111.00000000
Wildcard Mask	**0.0.0.255**	00000000.00000000.00000000.11111111
Result	192.168.1.0	11000000.10101000.00000001.00000000

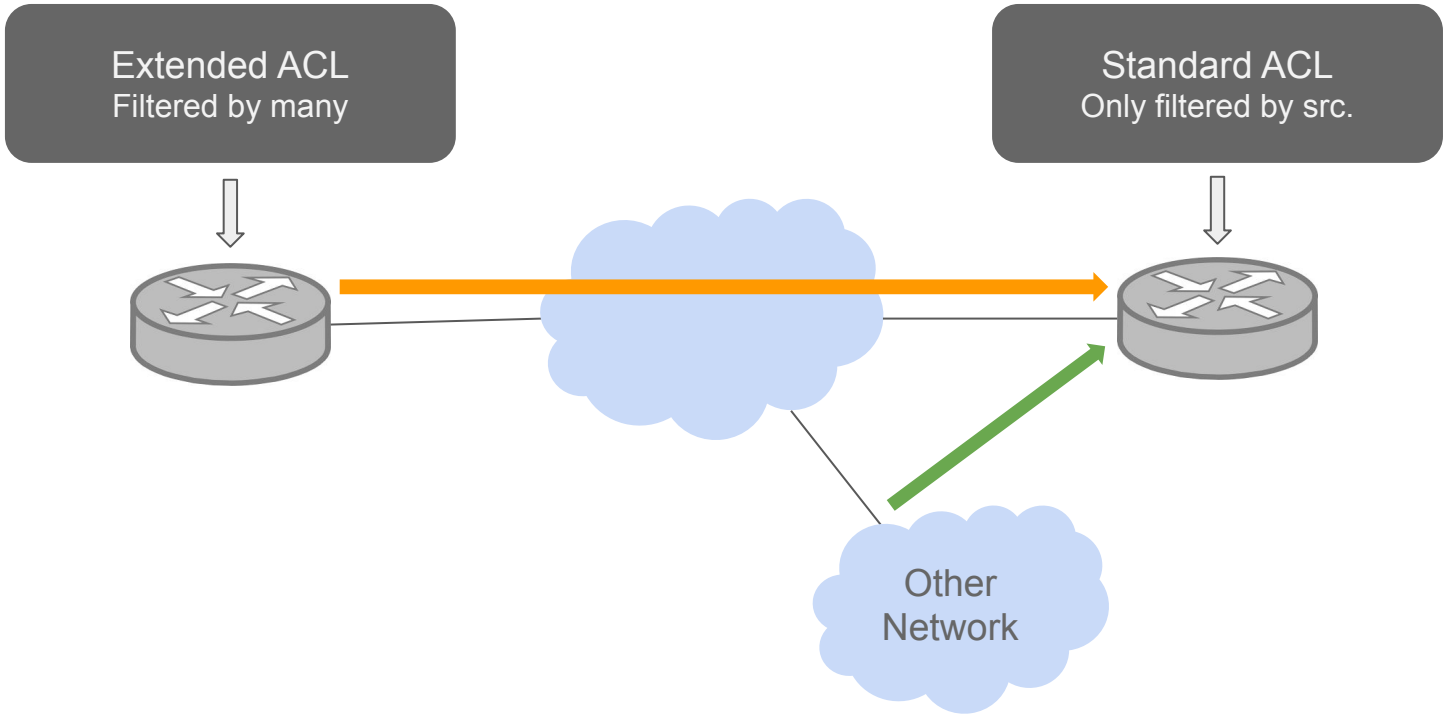
- Tips: Subtract the subnet mask from 255.255.255.255
  - 255.255.255.255 - 255.255.255.0 = 0.0.0.255

Access Control List (ACL) - Type

- Standard ACL
  - Only based on source IP address
  - Placed close to the destination due to the inability to specify destination address
- Extended ACL
  - Based on source IP, source port, destination IP, destination port and protocol
  - Placed close to the source to filter undesirable traffic

Access Control List (ACL) - Where to place

- Case: Deny traffic coming from several networks



Standard ACLs

- Can only check source IP address
- Numbered ACL
  - *list-number* should be 1-99, 1300-1999 (for Cisco IOS)

```
(config)# access-list list-number {deny|permit} source-ip [wildcard]
(config)# access-list list-number remark description
```

- Named ACL

```
(config)# ip access-list [standard|extended] list-name
(config-std-nacl)# [permit|deny|remark] source-ip [wildcard]
```

Standard ACLs - Example for Numbered ACL

```
Core(config)# access-list 20 permit 192.168.2.2
Core(config)# access-list 20 permit 192.168.2.3
Core(config)# exit
Core# show ip access-lists
Standard IP access list 20
    10 permit host 192.168.2.2
    20 permit host 192.168.2.3
Core# show ip access-lists 20
Standard IP access list 20
       permit host 192.168.2.2
       permit host 192.168.2.3
```

Standard ACLs - Example for Named ACL

```
Core(config)# ip access-list standard test-acl
Core(config-std-nacl)# permit 192.168.2.2
Core(config-std-nacl)# permit 192.168.2.3
Core(config-std-nacl)# exit
Core(config)# exit
Core# show ip access-lists
Standard IP access list test-acl
    10 permit host 192.168.2.2
    20 permit host 192.168.2.3
Core# show ip access-lists test-acl
Standard IP access list 20
       permit host 192.168.2.2
       permit host 192.168.2.3
```

Extended ACLs

- Having the ability to check src_IP, src_port, dst_IP, dst_port and protocol
- Numbered ACL
  - *list-number* should be 100-199 or 2000-2699 (std ACL: 1-99, 1300-1999)

```
(config)# access-list list_number {permit|deny} protocol src-ip wildcard
[operator {port-num|service}] dst-ip wildcard [operator {port-num|service}]
```

- Named ACL

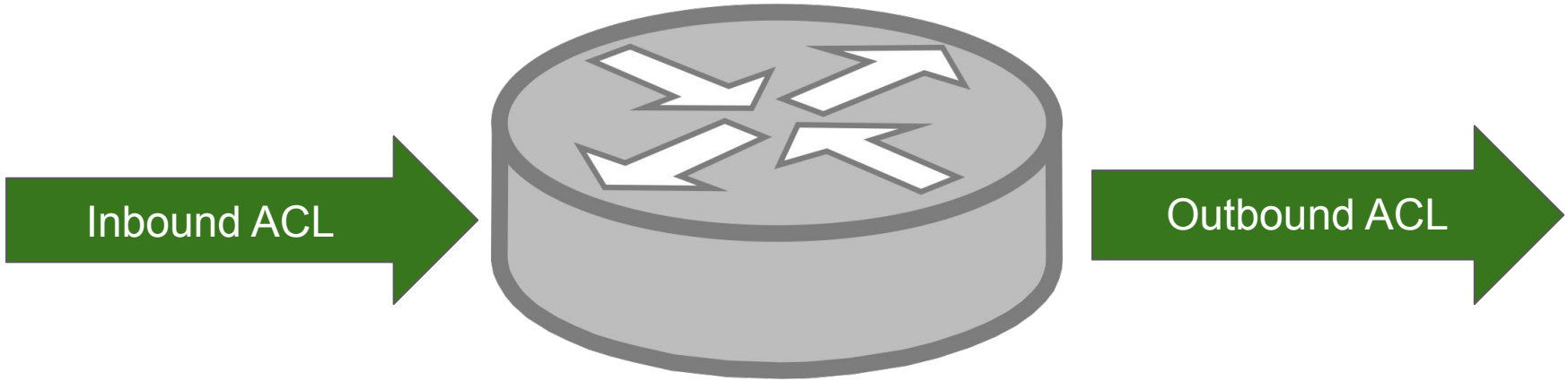
```
(config)# ip access-list extended list_name
(config-ext-nacl)# {permit|deny} protocol src-ip wildcard [operator
{port|service}] dst-ip wildcard [operator {port-num|service}]
```

Extended ACLs - Example

- Use named ACL for example

```
Core(config)# ip access-list extended PC-1-to-2
Core(config-ext-nacl)# permit ip host 192.168.1.2 host 192.168.2.2
Core(config-ext-nacl)# permit ip host 192.168.1.3 host 192.168.2.3
Core(config-ext-nacl)# ^Z
Core# show ip access-lists PC-1-to-2
Extended IP access list PC-1-to-2
    10 permit ip host 192.168.1.2 host 192.168.2.2
    20 permit ip host 192.168.1.3 host 192.168.2.3
```

Access Control List (ACL) - Flow Direction



- Inbound ACL
  - Filter packets coming to a specific interface, before routed to outbound interface
- Outbound ACL
  - Filter packets after being routed, regardless of the inbound interface

Configure (Apply) ACL on interface

- Apply to interface

```
(config-if)# ip access-group {list-number|list-name} {in|out}
```

- Apply to line (e.g. console, vty)

```
(config-line)# access-class {list-number|list-name} {in|out}
```

- **in**: Apply to incoming connections
- **out**: Apply to outgoing connections

Configure ACL on interface - Example

```
Core(config)# interface GigabitEthernet 0/1
Core(config-if)# ip access-group PC-1-to-2 in
```

- On PC 1-2, ping 192.168.2.2 & 192.168.2.3

```
C:\> ping 192.168.2.2
...
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127
Reply from 192.168.2.2: bytes=32 time<1ms TTL=127
...
C:\> ping 192.168.2.3
...
Reply from 192.168.2.3: Destination host unreachable.
Reply from 192.168.2.3: Destination host unreachable.
...
```

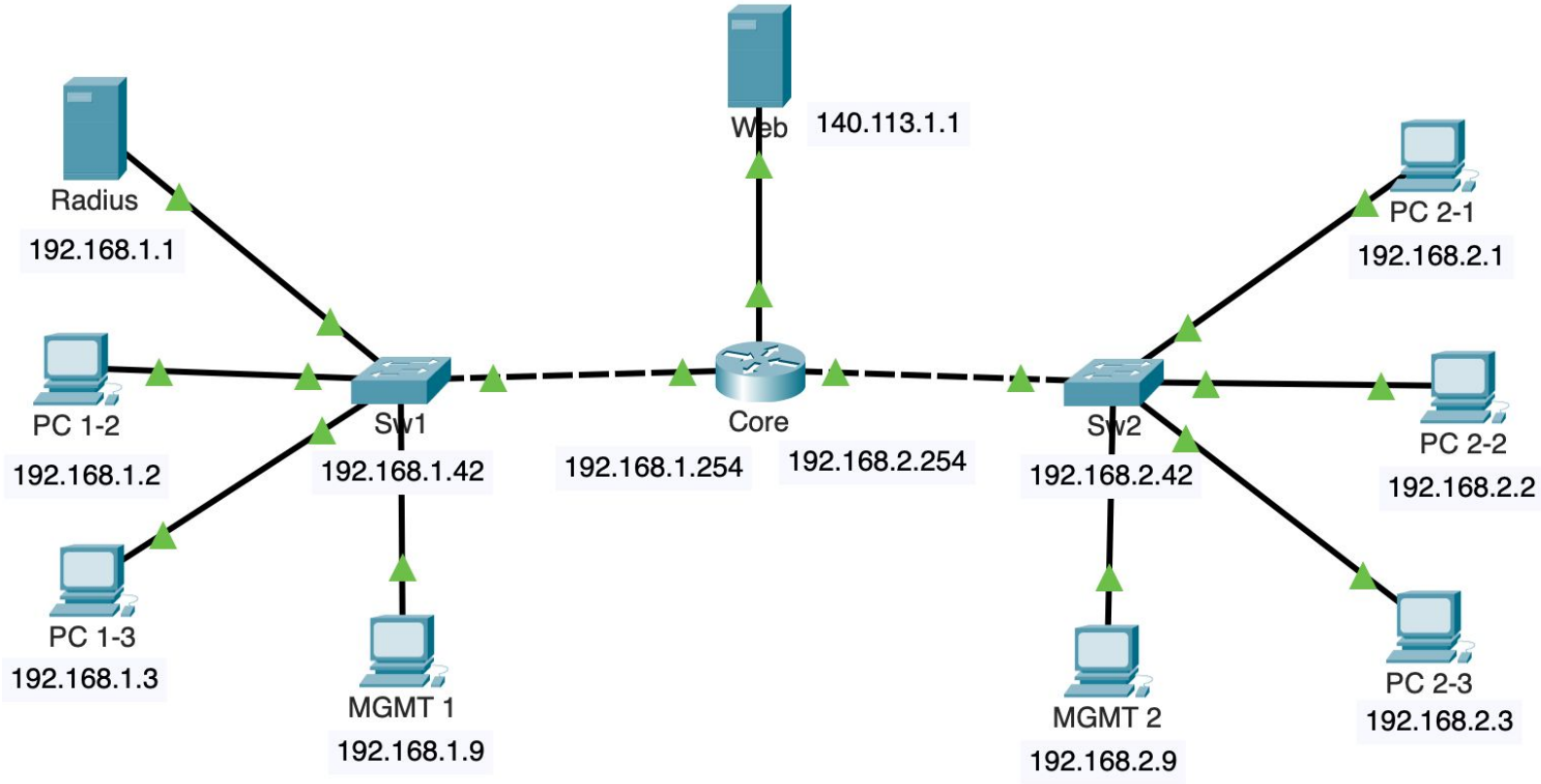
Sequence Numbers

- Every time you add a new entry in ACL, it is added at the end of it.
  - According to first match rule, that entry will be checked at last.
- How to insert an entry between previous ACL entries?
  - Use sequence numbers
- Sequence numbers help you make the proper order of ACL entries

Verify Setting - Example

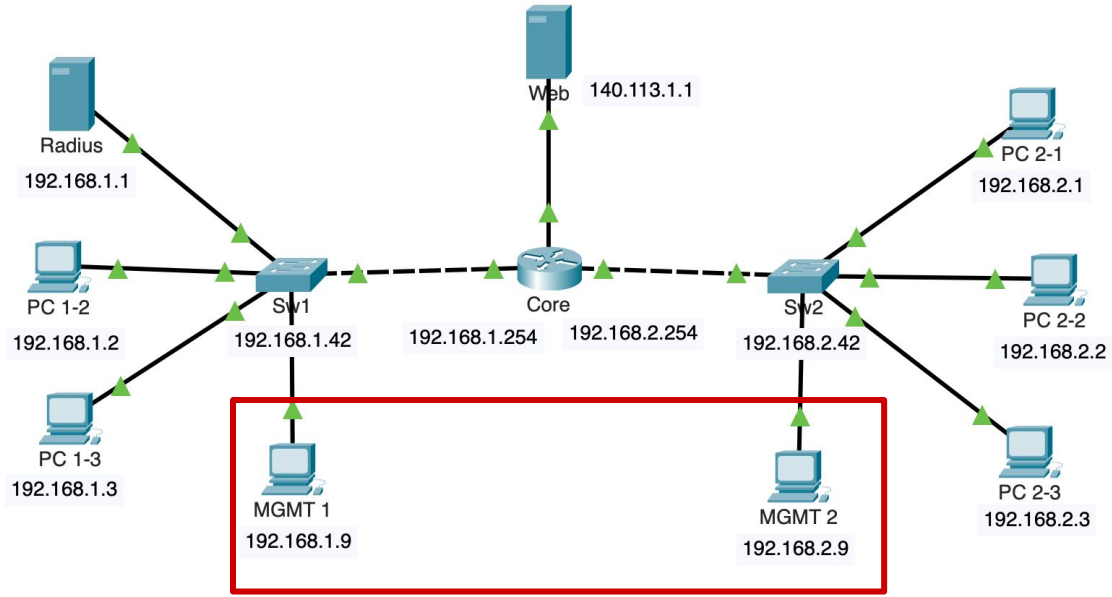
```
Core(config)# do show ip access-lists
...
Extended IP access list PC-1-to-2
    10 permit ip host 192.168.1.2 host 192.168.2.2
    20 permit ip host 192.168.1.3 host 192.168.2.3
...
Core(config)# ip access-list extended PC-1-to-2
Core(config-ext-nacl)# no 10
Core(config-ext-nacl)# 10 permit ip host 192.168.1.1 any  ! Radius
Core(config-ext-nacl)# do show ip access-lists
...
Extended IP access list PC-1-to-2
    10 permit ip host 192.168.1.1 any
    20 permit ip host 192.168.1.3 host 192.168.1.3
...
```

ACL Practice



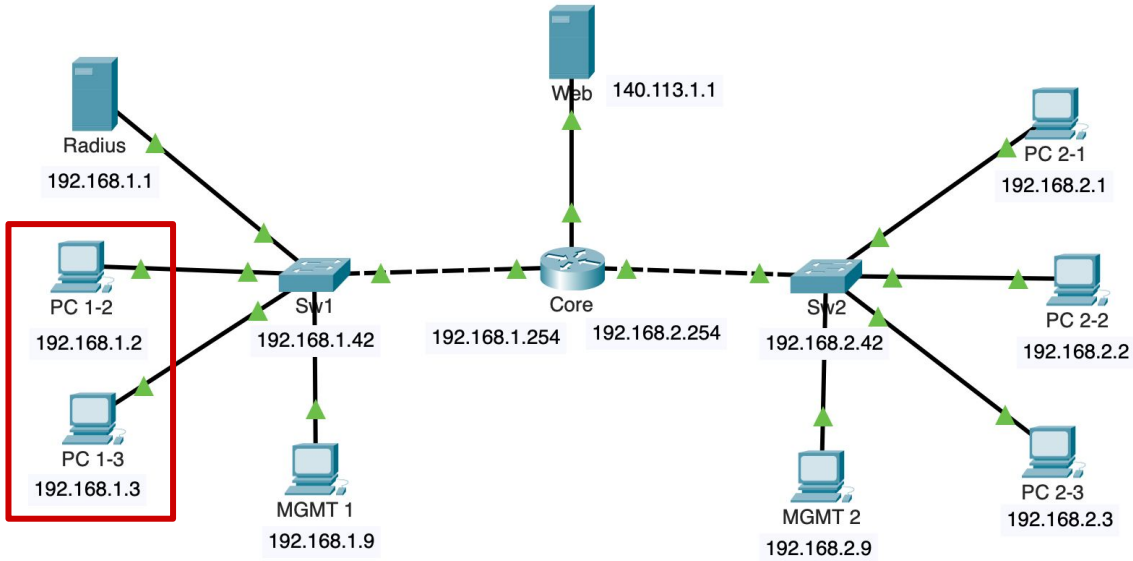
Requirements (Management)

- Management PC
  - Can ping all PC and servers (may not have pong)
  - Only ICMP is allowed



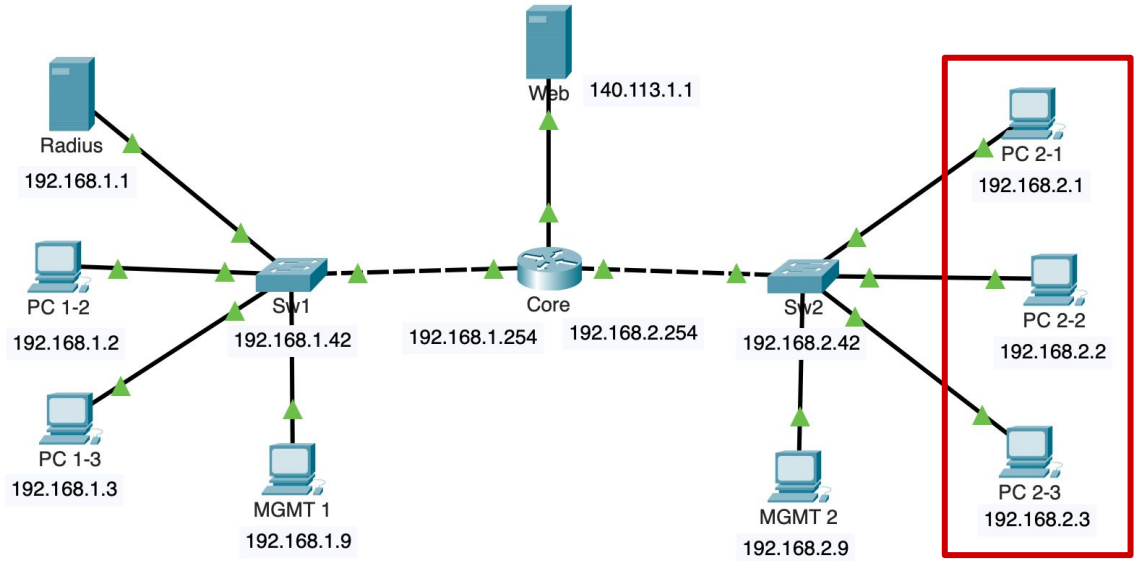
Requirements (Left)

- PC on the left side
  - Can only ping MGMT 1 (may not have pong)
  - Can access Web server



Requirements (Right)

- If source IP is odd, can ping PC1-* (may not have pong)
- If source IP is even, can access Web server



Thanks

Any questions?