以 Ubuntu 18.04 建置透通式防火牆

108.04.26 by 劉勇炫 v1.1

一、透通式防火牆概說

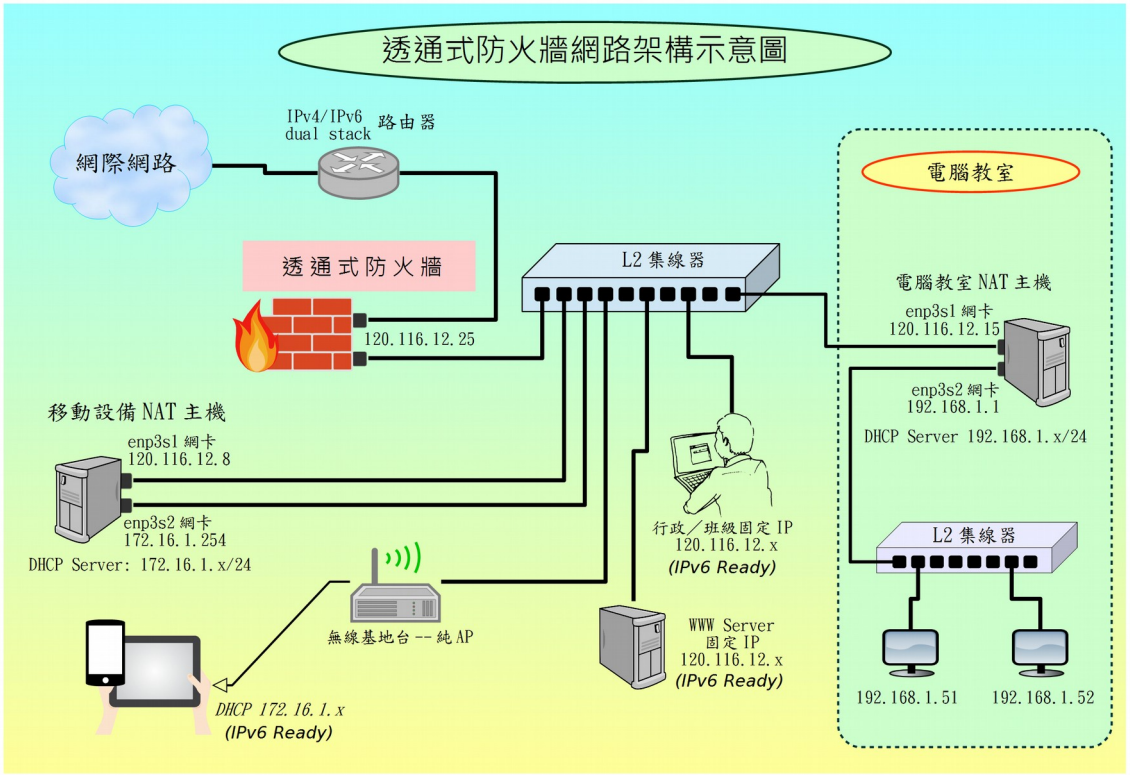
1. 運作概念

本文所介紹的透通式防火牆就是以一台主機硬體規格不必太高的 Linux 主機，安裝兩片伺服器專用網卡，一對內一對外，使其成為區網設備的「必經道路」，好方便我們可以攔截所有封包，進而達到管控的目的。

一旦此機制架設完畢，我們便可利用主機上的「Netfilter 工具」進行管控；「封包擷取工具」來監視。具體來說它可以做到：異常封包或異常流量偵測；網卡封鎖；禁行特定網站；流量限制； DoS 攻擊預防；通訊埠鎖定等。

當然這些作業，也可以用任何市售的防火牆硬體達成。但筆者推薦使用 Linux 來架設的理由有二：一來市面上能達成 100M 以上流量的硬體 firewall 通常索價不菲；二來，無論是觀察或控管，Linux 上已有豐富的工具供我們使用，商用機器則是每個功能都得加錢。

2. 運作方式



透通式防火牆架構示意圖

如圖 4-10 所示，它是架在區域網路的次上層：路由器之後，由一台電腦主機加上兩張網卡組成。在這台電腦裝上 Ubuntu Linux 作業系統，並使兩張網卡使用同一組 IP 位址。

二、透通式防火牆架設

在 ubuntu 18.04 下的架設過程大致如下：

- 準備一台主機內含兩張網卡（或單卡雙埠）
- 建立開機自動執行 rc.local 機制
- 安裝所需套件 bridge-utils

```
root@fw:~# apt install bridge-utils
```

- 用 lshw 指令查出網卡代號，本文以
  - enp2s0f0 為對外網卡
  - enp3s0 對內網卡
- 修改 /etc/rc.local，做好相關設定

(一). 開機自動執行 script 機制(rc-local)建立

1. rc-local.service 概述

在 DOS 時代，我們會把開機後立即執行的指令寫在 autoexec.bat 內，Linux 這樣的機制是在 /etc/rc.local 內。早期 Linux 普遍使用 sysvinit 來管理啟動程序，/etc/rc.local 就單純只是一個等候被執行的 SHELL，到了 systemd 的時代，它把 rc.local 這個機制變成了一個服務叫 rc-local.service，在 Ubuntu 18.04 這個服務已不存在，必須自行添加，才可以正常運作。方式如下：

A. 以 root 身份建立 /etc/systemd/system/rc-local.service 服務啟動程序

```
user@ubuntu:~$ sudo su
[sudo] password for user:
root@ubuntu:/home/user# cd
root@ubuntu:~# vi /etc/systemd/system/rc-local.service
```

輸入以下內容

```
[Unit]
Description=/etc/rc.local Compatibility
ConditionPathExists=/etc/rc.local

[Service]
Type=forking
ExecStart=/etc/rc.local start
TimeoutSec=0
StandardOutput=tty
RemainAfterExit=yes
SysVStartPriority=99

[Install]
WantedBy=multi-user.target
```

B. 建立 /etc/rc.local 檔案

```
root@ubuntu:~# vi /etc/rc.local
```

加入以下內容，因為筆者習慣使用 bash ，所以開頭第一行指定 /bin/bash

```
#!/bin/bash
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.


exit 0
```

C. 賦予 rc.local 執行權

```
root@ubuntu:~# chmod +x /etc/rc.local
```

建置完畢後，可在重開機後用「systemctl status rc-local」查一下 rc.local 是否被執行。

```
root@ubuntu:~# systemctl status rc-local
● rc-local.service - /etc/rc.local Compatibility
   Loaded: loaded (/lib/systemd/system/rc-local.service; static; vendor preset: enabled)
  Drop-In: /lib/systemd/system/rc-local.service.d
           └─debian.conf
   Active: active (exited) since 二 2016-12-20 16:24:20 CST; 1 weeks 4 days ago
  Process: 789 ExecStart=/etc/rc.local start (code=exited, status=0/SUCCESS)

Warning: Journal has been rotated since unit was started. Log output is incomplete or
unavailable.
```

由以上可看到，它目前的狀況是「啟動中；active（exited）」。不必再做任何事。

但若 Active: 狀態是「inactive（dead）」，那麼要把它啟用，並隨開機而啟動，方式如下：

```
root@ubuntu:~# systemctl enable rc-local
root@ubuntu:~# systemctl start rc-local
```

2. 用 iptables 指令測試

我們把防火牆指令寫入 rc.local ，並在重新開機後檢查一下那些規則是否被執行。

- Lubuntu 安裝完畢，預設是沒有任何 iptables 規則，檢查方式如下：

```
user@ubuntu:~$ sudo -i
[sudo] password for user:
root@ubuntu:~# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
```

```
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- 為 /etc/rc.local 加上本機 Port 22 的本機防禦規則

```
roor@ubuntu:~# vi /etc/rc.local
```

- 修改後的 /etc/rc.local 如下：

```
#!/bin/bash
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
IPTABLES="/sbin/iptables"
IP6TABLES="/sbin/ip6tables"
$IPTABLES -F
$IP6TABLES -F
###------------------------------------------------------###
# 設定 filter table 的預設政策
###------------------------------------------------------###
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT

###------------------------------------------------------###
# 設定 Port 22 規則
###------------------------------------------------------###
$IPTABLES -A INPUT -p tcp -s 120.116.12.0/23 --dport 22 -j ACCEPT
$IPTABLES -A INPUT -p tcp -s 127.0.0.1 --dport 22 -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 22 -j DROP

$IP6TABLES -A INPUT -p tcp -s 2001:288:75a6::/48 --dport 22 -j ACCEPT
$IP6TABLES -A INPUT -p tcp --dport 22 -j DROP

exit 0
```

- 馬上套用新的規則列

```
root@ubuntu:~# systemctl restart rc-local
```

- 檢視 IPv4 的 iptables 規則列是否已執行

```
root@ubuntu:~# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  120.116.12.0/23      0.0.0.0/0           tcp dpt:22
ACCEPT     tcp  --  127.0.0.1            0.0.0.0/0           tcp dpt:22
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- 檢視 IPv6 的 ip6tables 規則列是否已執行

```
root@ubuntu:~# ip6tables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp      2001:288:75a6::/48   ::/0                tcp dpt:22
DROP       tcp      ::/0                 ::/0                tcp dpt:22

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

筆者建置 rc.local 的過程，可簡述如下：

A. 建立 rc.local 服務啟動程序 rc-local.service
B. 編寫 rc.local
C. 執行「systemctl restart rc-local」套用
D. 檢視結果：本例是利用 iptables 規則列測試，所以用 iptables -nL 指令檢查
E. 執行無誤，重新開機（reboot）後再檢視一次

(二). 修改 rc.local 寫入相關設定值

上一段我們有用 iptables 規則列來測試 rc.local 是否被執行。在本文，/etc/rc.local 的內容必須先清空，改以下面文字改寫。修改的方向大致如下：

- 把兩張網卡綁成同一組 IPv4 及 IPv6 位址，本文範例：

  IPv4: 120.116.12.19/23

  IPv6: 2001:288:75a6::19/64
- 啟動內外網卡間 IPv4/IPv6 封包的 Forward 功能
- 設定 Linux 的 DNS Client: /etc/resolv.conf

  Ubuntu 18.04 的 DNS Client 設在 /run/systemd 底下，因此把相關參數改寫至這裡

```
#!/bin/bash

MODPROBE="/sbin/modprobe"
IFCONFIG="/sbin/ifconfig"
ROUTE="/sbin/route"
IPTABLES="/sbin/iptables"
IP6TABLES="/sbin/ip6tables"
# 請記得先安裝 bridge-utils 套件
BRCTL="/sbin/brctl"
SERVICE="/usr/sbin/service"

###-------------------------------------------------------###
# 設定網段 IP 及介面
# enp2s0fo -> 對外
# enp3s0 -> 對內
###-------------------------------------------------------###
# 不要忘了打開 FW_IFACE
BR_IP="120.116.12.19"
FW_IP="120.116.12.19"
FW_IP_RANGE="/23"
BR_BCAST="120.116.13.255"
BR_IFACE="br0"
GATEWAY="120.116.13.254"
FW_IP6="2001:288:75a6::19"
GATEWAY6="2001:288:75a6::fffe"

###-------------------------------------------------------###
# ubuntu 18.04 的 network-manager 仍是使用 sysvinit 的方式
# 所以仍是 service xxxx start/stop 架構
# 把 network-manager 關閉，才可自訂網卡參數，不然會被它搶走
###-------------------------------------------------------###
$SERVICE network-manager stop

# 先關閉所有網路介面
eth0exist=$(/sbin/ifconfig |grep "enp2s0f0")
if [ "$eth0exist" != "" ]; then
    $IFCONFIG enp2s0f0 down
fi
```

```
ethlexist=$(/sbin/ifconfig |grep "enp3s0")
if [ "$ethlexist" != "" ]; then
    $IFCONFIG enp3s0 down
fi

br0exist=$(/sbin/ifconfig |grep "br0")
if [ "$br0exist" != "" ]; then
    $IFCONFIG br0 down
    # 再關閉 birdge 的 binding
    $BRCTL delif br0 enp2s0f0
    $BRCTL delif br0 enp3s0
    $BRCTL delbr br0
fi

# 設定 enp2s0f0 及 enp3s0 網卡介面
$IFCONFIG enp2s0f0 0.0.0.0
$IFCONFIG enp3s0 0.0.0.0

#啟動 bridge 與實體網卡作 Binding
$BRCTL addbr br0
$BRCTL addif br0 enp2s0f0
$BRCTL addif br0 enp3s0

# 設定 br0 介面
$IFCONFIG br0 $BR_IP netmask 255.255.254.0 broadcast $BR_BCAST

# 設定 gateway 值
$ROUTE add default gw $GATEWAY

###-----------------------------------------------------###
# 打開 IPv4 的forward
###-----------------------------------------------------###
#echo "Enable ip_forward ......"
#echo
echo "1" > /proc/sys/net/ipv4/ip_forward

###-----------------------------------------------------###
# IPv6 透通
###-----------------------------------------------------###
ipv6addr=$($IFCONFIG |grep "inet6 addr: $FW_IP6")
if [ "$ipv6addr" == "" ]; then
    /bin/ip -6 addr add $FW_IP6/64 dev br0
fi
$ROUTE -A inet6 add ::/0 gw $GATEWAY6
echo "1" > /proc/sys/net/ipv6/conf/all/forwarding



###-----------------------------------------------------###
# 清除先前的設定
###-----------------------------------------------------###
echo "Flush fiter table ......"
```

```
echo

# Flush filter
$IPTABLES -F
$IPTABLES -X
$IP6TABLES -F

echo "Flush mangle table ......"
echo
# Flush mangle
$IPTABLES -F -t mangle
$IPTABLES -t mangle -X


echo "Flush nat table ......"
echo
# Flush nat
$IPTABLES -F -t nat
$IPTABLES -t nat -X


# 預設政策，全開
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT

$IP6TABLES -F

###-----------------------------------------------------###
# 本機防禦措施
###-----------------------------------------------------###
$IPTABLES -A INPUT -p tcp -s 120.116.12.0/23 --dport 22 -j ACCEPT
$IPTABLES -A INPUT -p tcp -s 127.0.0.1 --dport 22 -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 22 -j DROP

$IP6TABLES -A INPUT -p tcp -s 2001:288:75a6::/48 --dport 22 -j ACCEPT
$IP6TABLES -A INPUT -p tcp --dport 22 -j DROP

###-----------------------------------------------------###
# 阻擋某張網卡對外連線之範例寫法
###-----------------------------------------------------###
# $IPTABLES -A FORWARD -p tcp -m mac --mac-source 00:25:11:49:12:6C -j DROP

# dns client -> /etc/resolv.conf
echo "nameserver 168.95.1.1
nameserver 8.8.8.8" > /run/systemd/resolve/stub-resolv.conf

exit 0
```

設好了之後，先重新開機，依以下步驟測試：

A. 用瀏覽器測試本機是否可以上網

B. 把對內的線材接至集線器後，測試在內網的機器是否可以上網

三、DHCP SERVER 建置

若本 Gateway 身兼 DHCP SERVER，便得架設 isc-dhcp-server 過程如下。

1. 安裝 DHCP Server

用 root 身份以 apt 指令安裝 dhcp server

```
root@nat:~# apt update
root@nat:~# apt install isc-dhcp-server
```

2. 修改 /etc/default/isc-dhcp-server 指定派送 ip 的網卡代號

```
root@nat:~# vi /etc/default/isc-dhcp-server
```

透通式的網卡已被 rc.local 指定為「br0」，所以要把派送 IP 的網卡介面設成 br0，示例如下

```
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
# 預設只啟用 IPv4 位址的派送
INTERFACESv4="br0"
INTERFACESv6=""
```

3. 修改 /etc/dhcp/dhcpd.conf 自訂派送範圍

```
root@nat:~# vi /etc/dhcp/dhcpd.conf
```

依本文範例中粗體字部分進行修改，依本例：

A. 網域名稱：dces.tn.edu.tw

B. 派送時指定 dns server 為 168.95.1.1 及 8.8.8.8

C. 自動派送範圍 120.116.13.101 ～ 120.116.13.200 。

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
# option definitions common to all supported networks...
option domain-name "dces.tn.edu.tw";
option domain-name-servers 168.95.1.1, 8.8.8.8;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

subnet 120.116.12.0 netmask 255.255.254.0 {
  range 120.116.13.101 120.116.13.200;
  option routers 120.116.13.254;
  option broadcast-address 120.116.13.255;
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
```

```
#subnet 10.5.5.0 netmask 255.255.255.224 {
#  range 10.5.5.26 10.5.5.30;
#  option domain-name-servers ns1.internal.example.org;
#  option domain-name "internal.example.org";
#  option subnet-mask 255.255.255.224;
#  option routers 10.5.5.1;
#  option broadcast-address 10.5.5.31;
#  default-lease-time 600;
#  max-lease-time 7200;
#}

# Hosts which require special configuration options can be listed in
# host statements.   If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.

#host passacaglia {
#  hardware ethernet 0:0:c0:5d:bd:95;
#  filename "vmunix.passacaglia";
#  server-name "toccata.example.com";
#}

# Fixed IP addresses can also be specified for hosts.   These addresses
# should not also be listed as being available for dynamic assignment.
# Hosts for which fixed IP addresses have been specified can boot using
# BOOTP or DHCP.   Hosts for which no fixed address is specified can only
# be booted with DHCP, unless there is an address range on the subnet
# to which a BOOTP client is connected which has the dynamic-bootp flag
# set.
#host fantasia {
#  hardware ethernet 08:00:07:26:c0:a5;
#  fixed-address fantasia.example.com;
#}

# You can declare a class of clients and then do address allocation
# based on that.   The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and all
# other clients get addresses on the 10.0.29/24 subnet.

#class "foo" {
#  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
#}

#shared-network 224-29 {
#  subnet 10.17.224.0 netmask 255.255.255.0 {
#    option routers rtr-224.example.org;
#  }
#  subnet 10.0.29.0 netmask 255.255.255.0 {
#    option routers rtr-29.example.org;
#  }
#  pool {
#    allow members of "foo";
```

```
#    range 10.17.224.10 10.17.224.250;
#  }
#  pool {
#    deny members of "foo";
#    range 10.0.29.10 10.0.29.230;
#  }
#}
```

4. 啟用 DHCP 伺服器並檢查

- 手動重新啟動服務

```
root@nat:~# systemctl start isc-dhcp-server
```

- 用 netstat -nltup 檢查 DHCP 的 port 67 是否啟用

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address    Foreign Address    State     PID/Program name
tcp        0      0 127.0.0.53:53   0.0.0.0:*          LISTEN    520/systemd-resolve
tcp        0      0 0.0.0.0:22      0.0.0.0:*          LISTEN    674/sshd
tcp        0      0 127.0.0.1:631   0.0.0.0:*          LISTEN    1352/cupsd
tcp6       0      0 :::22           :::*               LISTEN    674/sshd
tcp6       0      0 ::1:631         :::*               LISTEN    1352/cupsd
udp        0      0 127.0.0.53:53   0.0.0.0:*                    520/systemd-resolve
udp        0      0 0.0.0.0:67      0.0.0.0:*                    11360/dhcpd
udp        0      0 0.0.0.0:5353    0.0.0.0:*                    610/avahi-daemon: r
udp        0      0 0.0.0.0:36093   0.0.0.0:*                    610/avahi-daemon: r
udp        0      0 0.0.0.0:56669   0.0.0.0:*                    11360/dhcpd
udp6       0      0 :::44860        :::*                         610/avahi-daemon: r
udp6       0      0 :::61792        :::*                         11360/dhcpd
udp6       0      0 :::5353         :::*                         610/avahi-daemon: r
```

- 用 systemctl 檢查服務狀態

```
root@pcrs:~# systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor
preset: enabled)
   Active: active (running) since Thu 2019-04-25 13:51:24 CST; 1min 11s ago
     Docs: man:dhcpd(8)
 Main PID: 11360 (dhcpd)
    Tasks: 1 (limit: 4515)
   CGroup: /system.slice/isc-dhcp-server.service
           └─11360 dhcpd -user dhcpd -group dhcpd -f -4 -pf
/run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf
```

5. 啟用 DHCP 後 /etc/rc.local 的配套措施

因為 rc-local 它的執行時間是「開機後」，也就是所有服務都已跑完的時候，因此，在前文 DHCP 伺服器啟動時，「br0」這張網卡是還不存在的。因此，我們不得讓 isc-dhcp-server 隨開機執行，取而代之的是在 rc.local 裡設妥 br0 網卡後再執行。修改後的 rc.local 如下(粗體字部分)。

```
#!/bin/bash

MODPROBE="/sbin/modprobe"
IFCONFIG="/sbin/ifconfig"
ROUTE="/sbin/route"
IPTABLES="/sbin/iptables"
IP6TABLES="/sbin/ip6tables"
# 請記得先安裝 bridge-utils 套件
BRCTL="/sbin/brctl"
SERVICE="/usr/sbin/service"
SYSTEMCTL="/bin/systemctl"

###-------------------------------------------------###
# 設定網段 IP 及介面
# enp2s0fo -> 對外
# enp3s0 -> 對內
###-------------------------------------------------###
# 不要忘了打開 FW_IFACE
BR_IP="120.116.12.19"
FW_IP="120.116.12.19"
FW_IP_RANGE="/23"
BR_BCAST="120.116.13.255"
BR_IFACE="br0"
GATEWAY="120.116.13.254"
FW_IP6="2001:288:75a6::19"
GATEWAY6="2001:288:75a6::fffe"

#先關閉 isc-dhcp-server
$SYSTEMCTL stop isc-dhcp-server

###-------------------------------------------------###
# ubuntu 18.04 的 network-manager 仍是使用 sysvinit 的方式
# 所以仍是 service xxxx start/stop 架構
# 把 network-manager 關閉，才可自訂網卡參數，不然會被它搶走
###-------------------------------------------------###
$SERVICE network-manager stop

# 先關閉所有網路介面
eth0exist=$(/sbin/ifconfig |grep "enp2s0f0")
if [ "$eth0exist" != "" ]; then
    $IFCONFIG enp2s0f0 down
fi

eth1exist=$(/sbin/ifconfig |grep "enp3s0")
if [ "$eth1exist" != "" ]; then
```

```
    $IFCONFIG enp3s0 down
fi

br0exist=$(/sbin/ifconfig |grep "br0")
if [ "$br0exist" != "" ]; then
    $IFCONFIG br0 down
    # 再關閉 birdge 的 binding
    $BRCTL delif br0 enp2s0f0
    $BRCTL delif br0 enp3s0
    $BRCTL delbr br0
fi

# 設定 enp2s0f0 及 enp3s0 網卡介面
$IFCONFIG enp2s0f0 0.0.0.0
$IFCONFIG enp3s0 0.0.0.0

#啟動 bridge 與實體網卡作 Binding
$BRCTL addbr br0
$BRCTL addif br0 enp2s0f0
$BRCTL addif br0 enp3s0

# 設定 br0 介面
$IFCONFIG br0 $BR_IP netmask 255.255.254.0 broadcast $BR_BCAST

# 設定 gateway 值
$ROUTE add default gw $GATEWAY

###-----------------------------------------------------###
# 打開 IPv4 的forward
###-----------------------------------------------------###
#echo "Enable ip_forward ......"
#echo
echo "1" > /proc/sys/net/ipv4/ip_forward

###-----------------------------------------------------###
# IPv6 透通
###-----------------------------------------------------###
ipv6addr=$($IFCONFIG |grep "inet6 addr: $FW_IP6")
if [ "$ipv6addr" == "" ]; then
    /bin/ip -6 addr add $FW_IP6/64 dev br0
fi
$ROUTE -A inet6 add ::/0 gw $GATEWAY6
echo "1" > /proc/sys/net/ipv6/conf/all/forwarding


###-----------------------------------------------------###
# 清除先前的設定
###-----------------------------------------------------###
echo "Flush fiter table ......"
echo

# Flush filter
```

```
$IPTABLES -F
$IPTABLES -X
$IP6TABLES -F

echo "Flush mangle table ......"
echo
# Flush mangle
$IPTABLES -F -t mangle
$IPTABLES -t mangle -X


echo "Flush nat table ......"
echo
# Flush nat
$IPTABLES -F -t nat
$IPTABLES -t nat -X


# 預設政策，全開
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT

$IP6TABLES -F

###-----------------------------------------------------###
# 本機防禦措施
###-----------------------------------------------------###
$IPTABLES -A INPUT -p tcp -s 120.116.12.0/23 --dport 22 -j ACCEPT
$IPTABLES -A INPUT -p tcp -s 127.0.0.1 --dport 22 -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 22 -j DROP

$IP6TABLES -A INPUT -p tcp -s 2001:288:75a6::/48 --dport 22 -j ACCEPT
$IP6TABLES -A INPUT -p tcp --dport 22 -j DROP

###-----------------------------------------------------###
# 阻擋某張網卡對外連線之範例寫法
###-----------------------------------------------------###
# $IPTABLES -A FORWARD -p tcp -m mac --mac-source 00:25:11:49:12:6C -j DROP

# dns client -> /etc/resolv.conf
echo "nameserver 168.95.1.1
nameserver 8.8.8.8" > /run/systemd/resolve/stub-resolv.conf

#有了 br0 網卡了, 再打開 isc-dhcp-server
$SYSTEMCTL start isc-dhcp-server

exit 0
```