# 第十四單元

# iptables 防火牆

## 1.實驗目的

架設 iptables 防火牆，並有能力管制網路上資料的進出。

## 2.實驗設備

- 安裝 Linux 系統之電腦
- Webmin(http://www.webmin.com)
- iptables(http://www.iptables.org)

## 3.背景資料網路安全簡介

近年來網路安全愈來愈受到重視，根據調查，每年美國因受網路攻擊的損失高達 100 億美元，每年入侵事件的回報次數也愈來愈多，下表是 cert 統計至 2002 年底為止入侵事件的次數。

| 年份 | 數量 |
|---|---|
| 1995 | 171 |
| 1996 | 345 |
| 1997 | 311 |
| 1998 | 262 |
| 1999 | 417 |
| 2000 | 1090 |
| 2001 | 2437 |
| 2002 | 4129 |

而 2003 年的前二季也高達了 1993 次之多，顯示在美國這種高網路化的國家，網路入侵的事件也是層出不窮，再加上近年來恐怖攻擊行動也有部份是由網路來進行資訊的傳送，或者是發動癱瘓型的攻擊行動，而造成全球經濟的大損失。有鑑於此，各國政府的資訊管理部門都開始重視資訊安全，美國 FBI 與美國電腦安全協會 CSI 曾經指出，包含大型的美國企業、財務機構、大學及政府單位等機構中，

90%的單位曾偵測到遭受安全侵害，而 70%遭到比病毒更嚴重的侵害，也就是說，幾乎一半的單位宣稱曾有網路入侵、攻擊的行為，而導致財務損失。

攻擊的手法各式各樣，包括內部的濫用及外部的攻擊。而攻擊的類型有：

- **掃瞄（Scanning）**：cracker（怪客）或 hacker（駭客）會在進行入侵時事先將情報搜集好，比如說使用 snmp 的 port 偵測工具，或是利用 ping 來測試等，當他們收集了愈多的情報，當然就愈有把握入侵使用者的系統。
- **阻絕服務（Denial of Service；DoS）**：這一類的攻擊幾乎都會造成伺服器的當機，使得主機無法提供服務，攻擊者可以利用持續傳送不完整的封包方式，讓伺服器無暇回應，自然其他的服務就受到影響，比如說利用郵件炸彈之類的。
- **分散式阻絕服務（Distributed denial of service；DDoS）**：和 DoS 類似，DoS 是由單台主機進行攻擊，而 DDoS 則是利用網路上被入侵的主機，對同一台服務器進行 DoS 攻擊，使得伺服器無法承受主機所能處理的程序而當機。
- **竊取（sniffing）**：這類攻擊通常都是從客戶端及伺服器端或是伺服器對伺服器的通訊中擷取資訊，攻擊者通常都會抓取 TCP/IP 協定的封包，在重組或解碼後，就可以得到使用者的密碼或者是重要的資訊等，一般市面上賣的 sniffer 或者是 fluke 之類的網管軟體即是其中一類，像 Linux 中的 snort 也有類似 sniffer 的功能，若是有心人士在伺服器或是區域網路中安裝此類的軟體，那麼網路就可能會出現資料被盜用的危機。
- **劫持（Hijacking）**：這是另一種兩個點在通訊中進行攻擊的手法，通常是兩個通訊中的連線，由攻擊的第三者對其中之一的機器進行攻擊，當其中之一的機器被癱瘓後，第三者就偽裝成為其中的一個機器，繼續與另一台機器進行通訊，如此一來即可接收另一台機器的資訊。
- **實體（Physical）**：實體部份很容易就達成，試想，假設公司的 MIS 人員疏忽了實體線路的重要性，那麼當公司的有心人士要竊取公司的重要資料，即可利用很多方式，比如說利用數據機傳送，或是在一個區段中接一個集線器出來盜取資訊（可用 VLAN 來防護），或者是直接到主機旁進行破壞，這些都是有可能發生的，所以資訊系統的實體管理不容忽視。
- **系統漏洞（Bug）**：沒有一個作業系統或程序是絕對沒有錯誤的，比如 Linux 上的 Sendmail 套件、Windows 的 IIS 套件等都是惡名昭彰，不管任何的系統一定都會有問題，而攻擊者就是利用這些漏洞來進行入侵，管理者應該時時注意漏洞的更新，減少被入侵的機會。
- **後門（back door）**：這類事件的發生多半是跟著系統漏洞而產生，有時也是程式開發人員在程式中遺留下特殊的密碼，可以直接進入系統取得管理者的權限，大部份的後門都是不為人知，一旦被發現後情況都很嚴重。
- **社交工程（Social engineering）**：可以靠欺騙的手段來進行，也可以利用職務之便來接近，這些都是利用社交工程來進行取得對方資料的手段。

當然，也有許多的防護方法：

- **防火牆（Firewall）：**防火牆是網路安全的第一道防線，很多組織都會在伺服器或是路由器上建立防火牆的機制，並設定哪些通訊可以通過、哪些不行。防火牆的目的在於關閉一些不必要的通訊，以減少被入侵的機會，但是它並不能避免所有的網路攻擊行為。防火牆可以分成三種：封包過濾（packet-filtering firewall）、應用層防火牆（application firewall）與線路層防火牆（circuit-level firewall）。現今在企業中主要的應用為 NAT（Network Address Translatino；網路位址轉換）、VPN（Virtual Private Network；虛擬私有網路）及 Proxy（代理伺服器）。
- **封包過濾（Filter）：**封包過濾實作在 OSI 的網路層中，可以使用多種方法來過濾進入或者是離開的封包，如果是使用 IP 的話，可以根據 IP 標頭的許多欄位，如範例來源位址、目的地址、來源埠號、目的埠號等指標來進行封包過濾，也可以針對各個來源進行限制，或是限制本地端的某個位址和外面的網路進行接觸。不過，由於使用封包過濾會讓系統的負載加重，進而影響網路的品質，所以管理者必須適當的使用。
- **入侵偵測（Intrusion Detection System；IDS）：**前者介紹的防火牆及封包過濾，只是針對連線的部份進行限制或者是放行的動作，假設今天有個合法的連線進入了我們的系統，該如何確定這是個安全的連線呢？此時可以利用 IDS 來進行確認，IDS 提供了許多的偵測規則，當系統管理者啟動後，IDS 會根據連線的狀態比對規則表，假設發生不合法或是可疑的連線時會把它記錄下來，或是直接將這個連線中斷。換句話說，防火牆及封包過濾只是網路介面的大門而已，IDS 則是負責管理所有進出連線的警衛。
- **系統更新（Update）：**目前所有的安全危機，絕大部份都是因為系統出現了許多的漏洞而被攻擊者趁虛而入，網路安全機構每天都會發出許多套件的安全警告，但只有少數的管理者會真的去進行系統升級的工作，因而導致許多伺服器或是主機淪為攻擊者的跳板，有鑑於此，許多作業系統維護廠商都推出了系統更新的機制，希望能夠有效的降低因為沒有修正系統漏洞而發生的網路危機。不過，因為攻擊手法實在改變太快，更新的動作總是比較慢一步，想要確保自身主機的安全，還是得靠其他安全機制的配合，才能更有效的防護主機的正常運作。
- **憑證（Certificate）：**網際網路上的電子商務必需要提供安全的網路交易環境，而這些機制需要機密性、身份驗證特性及不可否認性，此時就需要一個可靠的第三方機構來驗證，而 CA（Certification Authority）就是專門來提供這種服務的機構，憑證機制是目前最常使用的一種安全認証機制。
- **資料加密（Encryption）：**在資料進行傳送或是點對點的過程中，所有的資料都是在網路中傳送，如果有第三者在你的網路中搜集資訊，這些以明文方式傳送的資料很可能就會被竊取成功，如早期的 telnet 服務、ftp 服務等。目前 ssh 已經取代了 telnet 的連線，而 ftp、pop3 或者是 smtp 也提供了加密的

機制，以確保資料在傳輸過程中不會使用明文的方式。

- **實體安全（Physical Security）：**機房的門禁管制、公司使用者的管理、不明人士的行為...等，這些都是實體安全要注意的地方，尤其在現今這個電子資訊化的社會，連一隻手機都可以造成企業極大的損失，所以資訊管理人員必須正視目前所處環境的安全問題。

網際網路的成長，大量且快速的延伸到商業結構與個人的生活中，遠遠的超過了大部份的組織或公司所以保護資訊安全的能力，許多的公司、組織不斷的增加網路設備，卻往往忽略了資訊安全的重要性，而個人使用者也幾乎忽略了自身主機的安全，才會造成每年都會有如此嚴重的損失情形，雖然沒有一個產品能夠完整的防護系統不被侵襲，擁有正確的防護知識一定可以幫助我們將損失降到最低。

防火牆簡介

防火牆基本上是為了預防別人來存取你的網路，進而管制網路上資料的進出。防火牆一端連接外部的網路（經由真實的 IP），另一端則連接內部的網路（虛擬的 IP），將內部的網路與外部的網路隔離開，防火牆成了進入內部網路的唯一通道，因此任何進出的資料都要經過防火牆，再經由防火牆來決定是否能夠通行。

**防火牆的種類**

- **封包過濾器：**這個功能是取得每一個封包經由所設定的規則去進行過濾，檢查是否允許封包的傳送或是拒絕封包；封包過濾器存在於網路階層，而且不會影響到封包的資料。在 Fedora Linux 中有一個 iptables 的套件（6.0 以上已內含），可以經由它來做封包過濾器。
- **代理伺服器（Proxy firewalls）：**代理伺服器通常又稱為應用程式閘道，允許通過防火牆間接進入網際網路。

## 4.實驗方法

轉換虛擬 IP

由於網際網路的發展愈來愈蓬勃，電腦的數量也急遽增加，導致目前 IP 位址難求，因此才出現了虛擬 IP 的解決方法。網路上保留了特定 IP 供給私人虛擬網路使用，在真實的網路上將不會找到這三組 IP，其虛擬 IP 位址如下圖所示。

| | | | |
|---|---|---|---|
| Class A | 10.0.0.0 | ~ | 10.255.255.255 |
| Class B | 172.16.0.0 | ~ | 172.31.255.255 |
| Class C | 192.168.0.0 | ~ | 192.168.255.255 |

**查看網路卡狀態**

首先必須要有兩張網路卡介面，一張對外（使用真實 IP）eth0，一張對內（使用虛擬 IP）eth0，執行 ifconfig 會出現網路卡的設定值，檢查兩張網路卡是不是抓到了。

在這裡要注意的是，抓到的 eth0 和 eth1 的設定值可能是相反的，也就是說 eth0 對應到的是真實的 IP、eth1 對應到的是虛擬 IP，如果是這種情況，就必須做修改，否則網路會連不出去。

執行 ifconfig 查看目前所啟動的網路卡界面，下圖是全部設好的狀態畫面。

```
[root@net122 network-scripts]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:02:44:13:46:E6
          inet addr:192.192.73.122  Bcast:192.192.73.127  Mask:255.255.255.128
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:406 errors:0 dropped:0 overruns:0 frame:0
          TX packets:251 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34438 (33.6 Kb)  TX bytes:43341 (42.3 Kb)
          Interrupt:5 Base address:0xc000

eth1      Link encap:Ethernet  HWaddr 00:48:54:5D:00:4B
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:105 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11271 (11.0 Kb)  TX bytes:60 (60.0 b)
          Interrupt:10 Base address:0xe000
```

如果只抓到一張網路卡，或是後來才安裝上去，就必須手動安裝另一張網路卡。首先切換目錄到/etc/sysconfig 中，找到 network 檔案，其內容為：

```
NETWORKING=yes
HOSTNAME=net122.ee.oit.edu.tw
GATEWAY=192.192.73.126
```

接著到/etc/sysconfig/network-scripts 目錄中，會看到如下圖所示的檔案。

```
[root@net122 network-scripts]#  ls
ifcfg-eth0          ifdown-isdn    ifup-ipsec   ifup-routes
ifcfg-eth1          ifdown-post    ifup-ipv6    ifup-sit
ifcfg-lo            ifdown-ppp     ifup-ipx     ifup-sl
ifdown              ifdown-sit     ifup-isdn    ifup-wireless
ifdown-aliases      ifdown-sl      ifup-plip    init.ipv6-global
ifdown-ippp         ifup           ifup-plusb   network-functions
ifdown-ipsec        ifup-aliases   ifup-post    network-functions-ipv6
ifdown-ipv6         ifup-ippp      ifup-ppp
```

目前要注意的是 ifcfg-eth0、ifcfg-eth1 這兩個檔案，如果在開始過程中的硬體偵測沒有設定好的話，就不會產生 ifcfg-eth1 檔案。首先將 ifcfg-eth0 複製成 ifcfg-eth1，執行 cp ifcfg-eth0 ifcfg-eth1 再來修改，其中 ifcfg-eth0 為對外網路卡

的設定檔，依自己的設備去修改。

```
[root@net122 network-scripts]# cat ifcfg-eth0
# Realtek|RTL-8139/8139C/8139C+
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.192.73.127
HWADDR=00:02:44:13:46:E6
IPADDR=192.192.73.122
NETMASK=255.255.255.128
NETWORK=192.192.73.0
ONBOOT=yes
TYPE=Ethernet
```

說明：

- 第一行指定網路卡的界面為：eth0。
- 第二行指定使用固定的 IP 位址：static。
- 第三行指定廣播：192.192.73.127。
- 第四行指定網卡硬體位址：00:02:44:13:46:E6。
- 第五行指定 IP 位址為：192.192.73.122。
- 第六行指定網路遮罩為：255.255.255.128。
- 第七行指定網路號為：192.192.73.0。
- 第八行指定是否在開機時啟動：yes。
- 第九行指定種類為：Ethernet。

接著直接修改設定檔 ifcfg-eth1，做為內部虛擬的網路卡介面，所以真正可以用的虛擬 ip 位址為 192.168.1.1 ~ 192.168.1.254。

```
[root@net122 network-scripts]# cat ifcfg-eth1
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.1.254
NETMASK=255.255.255.0
HWADDR=00:48:54:5D:00:4B
```

說明：

- 第一行指定網路卡的界面為：eth1。
- 第二行指定在開機時啟動：yes。
- 第三行指定為靜態 IP 位址：static。
- 第四行指定 IP 位址為：192.168.1.254。
- 第五行指定網路遮罩為：255.255.255.0。
- 第六行指定網卡硬體位址：00:48:54:5D:00:4B。

**啟動網路卡**

可以個別對網路卡做啟動，如下表所示。

| | 啟動 | 關閉 |
|---|---|---|

| | | |
|---|---|---|
| 執行 | ifconfig eth0 up | ifconfig eth0 down |
| 執行 | ifconfig eth1 up | ifconfig eth1 down |

或是執行/etc/init.d/network restart，重新啟動整個網路，如下：

```
[root@net122 network-scripts]# /etc/rc.d/init.d/network restart
Shutting down interface eth0:          [    OK    ]
Shutting down interface eth1:          [    OK    ]
Shutting down loopback interface:      [    OK    ]
Setting network parameters:            [    OK    ]
Bringing up loopback interface:        [    OK    ]
Bringing up interface eth0:            [    OK    ]
Bringing up interface eth1:            [    OK    ]
[root@net122 network-scripts]#
```

**設定路由表**

當上述的檔案設定完、啟動之後，這兩個網路便會去建立預設的 route（路由）。其路由表為：route。

```
[root@net122 network-scripts]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.192.73.0    *               255.255.255.128 U     0      0        0 eth0
192.168.1.0     *               255.255.255.0   U     0      0        0 eth1
169.254.0.0     *               255.255.0.0     U     0      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         192.192.73.126  0.0.0.0         UG    0      0        0 eth0
[root@net122 network-scripts]#
```

其中 route 的命令為：

```
route add -net network address netmask device
```

| 網域 | **虛擬網域** |
|---|---|
| 網路號（network） | 192.168.1.0 |
| 網路遮罩（netmask） | 255.255.255.0 |
| 閘道（gateway） | 192.168.1.1 |

預設路由的設法：

```
route add default gw 192.192.73.126
```

替虛擬網域設定路由：

```
route add –net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.254
```

進行到這即安裝好兩張網路卡，eth0 就做為對外部的網路卡（真實 IP），eth1 做為對內部的網路卡（虛擬 IP）。

如果要刪除掉虛擬網域路由，只要將 add 改成 del 再執行一次即可：
route del –net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.254

**啟動 NAT 功能**

以 eth1 做為對內的網路介面，其虛擬 IP 位址為 192.168.1.0 ~ 192.168.1.255，第一個是網路號碼、最後一個是廣播號，所以可用的虛擬 IP 為 192.168.1.1 ~ 192.168.1.254，將閘道器（gateway）設為 192.168.1.254、子網路遮罩設為 255.255.255.0，將 192.168.1.1 ~ 192.168.1.253 之間的 IP 分配給內部的機器，之後內部的機器就可以互相通訊（ping），但對於要連出去還需要一個步驟，就是使用 iptables 程式來達成這個目的。先針對上述的問題，如果要讓內部的機器連接到外部的網路，請執行：

**echo 1 > /proc/sys/net/ipv4/ip_forward**
**iptables –t nat –A POSTROUTING -s 192.168.1.0/24 -j MASQUERADE**

說明：

- 第一個命令是將 ip_forward 轉送的功能打開。
- 第二個命令會將來源 192.168.1.0 ~ 192.168.1.255 的封包使用 IP 偽裝，將偽裝的封包送轉送給預設的路由到外部的網路，如此一來 NAT 的功能就算打開了，只要將第二張網路卡接到集線器上，其餘的機器也接到集線器後，就能透過虛擬 IP 連線出去。

可以將這兩行命令加在/etc/rc.d/rc.local 檔案中，使其每次開機時執行。

由於 Linux Kernel 版本的改進，使得封包過濾程式也跟著改變，在 Kernel 2.2.x 之前使用的是”ipchains”，Kernel 2.4.x 之後的版本改用"iptables”，在 Kernel 2.4 以後使用 netfilter 過濾機制，相關訊息可以在 netfilter 網站上（http://www.netfilter.org）上找到。

iptables 的主要特色有：

- 增強 NAT 功能
- 增強的封包檢視
- MAC 位址過濾
- 一定比例的限制條件
- 服務的優先順序類型

iptables 的語法

在此是以 Fedora Linux Kernel，iptables 1.2.8 為例。利用 lsmod |grep ip 列出 iptables 相關的模組，如果系統沒有載入模組，請執行 modprobe 模組名稱。

```
[root@net122 network-scripts]# lsmod |grep ip
ipt_MASQUERADE          2296   1  (autoclean)
iptable_nat            21784   1  (autoclean) [ipt_MASQUERADE]
ip_conntrack           29256   1  (autoclean) [ipt_MASQUERADE iptable_nat]
iptable_filter          2444   0  (autoclean) (unused)
ip_tables              15776   5  [ipt_MASQUERADE iptable_nat iptable_filter]
[root@net122 network-scripts]#
```

我們可以利用 iptables 程式去設定、修改及查看在系統中所設的封包過濾規則，語法規則要設的簡單或複雜，端看對安全性的重視。

範例：

iptables -P INPUT DROP

iptables -t filter INPUT -i eth0 –s 0/0 –j DROP

說明：以上兩個範例都是不讓外部網路連到主機上，而其中的差異會在稍後做說明。

語法：

iptables [-t TABLES] -action chains [PATTERN] [-j TARGET] [--target 選項]

有各種的組合方式，多看幾個例子就大約可以知道。

| iptables 語法： | | | | | |
|---|---|---|---|---|---|
| iptables | -t 過濾表 (tables) | -規則鏈的命令 (action) | 規則鏈 (chains) | 比對條件 (pattern) | -j 目標 (target) |
| 範例： | iptables –t nat -A POSTROUTING -s 192.168.1.0/24 -j MASQUERADE | | | | |
| 說明：指定 table 做 nat 功能，-A 新增一條 POSTROUTING 規則鏈，以 192.168.1.0/24 做來源位址，-j MASQUERADE 做 IP 偽裝 | | | | | |

iptables 的語法表示

iptables 的過濾表

語法：

iptables [-t TABLES] -action chains [PATTERN] [-j TARGET] [--target 選項]

有三種 tables（過濾表）可以處理：filter、nat 與 mangle，每個 tables 又有不同的

chains（規則鏈）可以去做對應處理，若在執行 iptables 時不加執行某一種 tables，則內定為”filter” table。

| tables(過濾表) | Chains(規則鏈) 大寫 |
|---|---|
| filter(過濾)(預設的 tables) | INPUT |
| | FORWARD |
| | OUTPUT |
| nat(網路位址轉換) | PREROUTING |
| | POSTROUTING |
| | OUTPUT |
| mangle(改變封包的資訊(TOS))<br>kernel2.4.18 後有 5 種<br>chains | PREROUTING |
| | INTPUT |
| | FORWARD |
| | OUTPUT |
| | POSTROUTING |

範例：

```
iptables –A OUTPUT –s 192.192.73.122 -d 0.0.0.0/24 -j ACCEPT
iptables –t nat -A POSTROUTING –o eth0 -j MASQUERADE
```

說明：

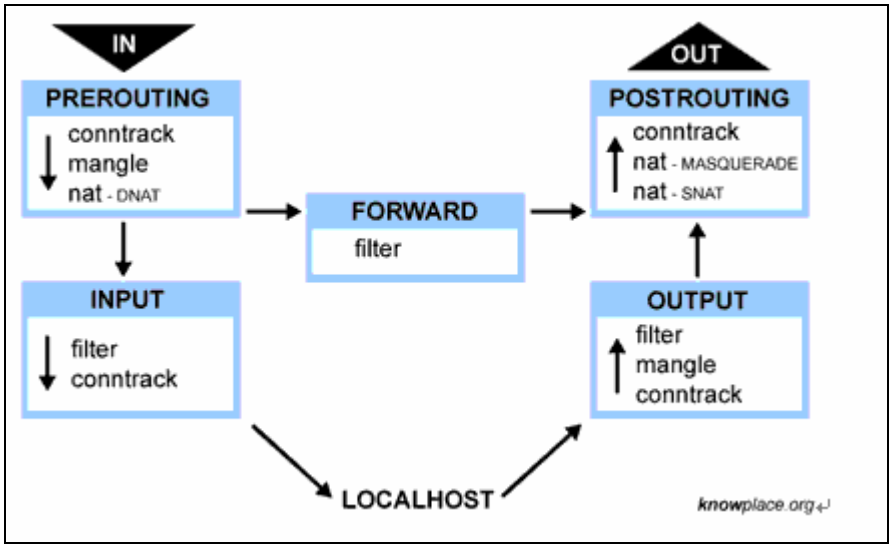
- 第一行是不指定 tables，所以內定為”filter”，而後面的 chains 只能用 INPUT、FORWARD、OUTPUT 這三種，讓 192.192.73.122 的封包出去。
- 第二行是指定 nat table，做以 eth0 為輸出、ip 偽裝。

iptables 的規則鏈

規則鏈就是能讓封包傳輸的規則。

語法：

```
iptables [-t TABLES] -action chains [PATTERN] [-j TARGET] [--target 選項]
```




說明：

- **PREROUTING**：在做 ROUTING 前尚未交給路由判斷之前的處理，這裡對進入的封包做目的地的改變（DNAT），可以使封包轉址到虛擬 ip 上。
- **INPUT**：當外部網路的封包要進入防火牆主機時進行過濾封包。
- **FORWARD**：轉送內部網路與外部網路的封包，封包不經過防火牆主機。
- **POSTROUTING**：在路由判斷之後，通常是封包離開界面之前的處理，在此進行 SNAT 可以使虛擬 ip 轉址連線出去。
- **OUTPUT**：當本地主機要送出封包時去過濾封包。

若想要更詳細的說明整個封包傳輸過程，可以連上 http://www.knowplace.org/netfilter/ipt_flow_mirror.html 查詢。

iptables 的命令

語法：

```
iptables [-t TABLES] -action chains [PATTERN] [-j TARGET] [--target 選項]
```

- 「**--list –L**」：可查看選擇設定的 iptables 規則，如果沒有規則被指定，會列出所有的規則（在-L 後可加-n 為取消 DN 查詢，直接使用 IP 方式顯示，速度會比較快）。

範例 1：iptables –L

說明：列出 filter tables（預設的 tables）中所設的規則，其有三條規則鏈：INPUT、FORWARD 及 OUTPUT。

```
[root@net122 root]# iptables -L
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
[root@net122 root]#
```

範例 2：iptables –t nat –L

說明：列出 nat table 中所設的規則，其中已有一條做 ip 偽裝的規則在。

```
[root@net122 root]# iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  192.168.1.0/24       anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@net122 root]#
```

■ 「**--append -A**」：可以增加一條新的規則鏈。

範例：

iptables -A OUTPUT –p tcp –s 192.192.73.122 –d 0.0.0.0/24 --dport 23 –j ACCEPT

說明：新增一條使 192.192.73.122 主機可以 telnet 到外部網路的規則。

■ 「**--delete -D**」：刪除 iptables 規則有兩種方法，第一種是知道在 OUTPUT 中的第 n 條規則，並指定刪除即可。第二種跟增加新的規則差不多，只不過是-A 指成-D，這種方法在設定很多的規則時候很好用，不必數它到底是第幾個，只要照條件去打即可。

範例：

iptables -D OUTPUT 1

iptables -D OUTPUT –p tcp –s 192.192.73.122 –d 0.0.0.0/24 --dport 23 –j ACCEPT

iptables –t nat – D POSTROUTING 1

說明：刪除在 OUTPUT 中的第一條規則，及刪除 nat table 的 POSTROUTING 中

的第一條規則。

■ 「**--insert -I**」：插入一個新 iptables 規則，其插入需指定規則中的數字，如果數字為 1，表示為第一個。

範例：
iptables -I OUTPUT 1 -s 192.192.73.122 -d 0.0.0.0/24 –j ACCEPT

說明：插入新的規則到 OUTPUT 的第 1 個位置。

■ 「**--replace -R**」：取代所選擇的規則，其取代需指定規則中的數字，跟 -I 一樣。

■ 「**--flush -F**」：將某個 iptables 規則清除，相當於刪除掉規則的功效，若沒指定規則，則會整個刪除掉。

範例：
iptables -F -t nat ； iptables -F

說明：清除 nat tables 中的所有規則；清除在 filter table 中的所有規則。

■ 「**--zero -Z**」：將所有規則中的封包和位元組計數歸零，它也可以去指定-L, --list (list）選項，會先列出之前的資料，再列出歸零的資料。

範例：
iptables -t nat -L -v -Z ; iptables –t nat -L -v

說明：列出所有 nat table 的規則，將封包、位元組清除歸零後，再列出規則，並加上-v 成為完整模式，然後查看封包數。

■ 「**--new-chain -N**」：產生一個新的使用者定義規則（user-defined）。

範例：
iptables -N TEST

說明：新增一個 TEST 的規則鏈。

■ 「**--delete-chain -X**」：只刪除使用者定義的規則，如果沒有指定任何的

參數，它將會刪除所有定義的規則。

範例：
iptables -t nat -X ； iptables -X

說明：清除在 nat tables 中使用者定義的規則；清除所有使用者定義的規則。

■ 「**--policy -P**」：設定 iptables 的預設政策，為 ACCEPT 和 DROP。

範例：
iptables -P INPUT DROP
iptables -p OUTPUT DROP
iptables -p FORWARD DROP

說明：將所有預設政策設為 DROP，使對內、對外的封包都中斷拒絕。

■ 「**--rename-chain -E**」：更改使用者定義的名稱。

範例：
iptables -N TEST
iptables -E TEST TEST2

說明：第一行為新增規則鏈，第二行為將 TEST 換成 TEST2。

■ 「**--help -h**」：列出描述命令語法的說明。

iptables 的規則比對選項
語法：
iptables [-t TABLES] -action chains [PATTERN] [-j TARGET] [--target 選項]

指定來源位址、目的位址可以有兩種形式：全名或縮寫來指定，使用'!'去反向，驚嘆號'!'有'not'的意義，很多選項都可以加上'!'來使用，表示不相等的意思。

範例：
-s ! localhost

說明：表示除了 localhost 的來源位址都可以。指定過濾對象-s -d 指定來源和目的地的 IP 位址。來源（-s）和目的地（-d）的表示法有 3 種：

1. 使用完整的主稱名稱，範例：’mouse.oit.edu.tw’ 或‘localhost’。
2. 使用 IP 位址，範例：’192.192.73.122’。
3. 一定範圍的 IP 位址，範例：’192.192.73.0/24’或’192.192.73.0/255.255.255.0’，兩者相同，都是包含 192.192.73.0 ~ 192.192.73.255 的 IP 位址。

斜線（‘/’）的數字代表 IP 位址，’/24’是 255.255.255.0，’/32’是 255.255.255.255，其中比較重要的是’0/0’，指全部。

-- source -s [!]address[/mask]
指定來源位址（ip/hostname），可以指定一段來源 address/mask。

範例：
iptables –A INPUT –s www.tslg.idv.tw -p icmp --icmp-type 8 -j DROP

說明：拒絕來自 www.tslg.idv.tw 的 icmp 協定的封包（ping），直接丟棄。

--destination -d [!]address[/mask]
指定目的地位址。

範例：
iptables –A OUTPUT -s 192.192.73.122 –d 0/0 -j ACCEPT

說明：讓 192.192.73.122 這台主機可以連出去。

**指定協定種類**
參數簡寫敘述：
--proto -p [!] protocol

協定可以用數字或名字來表示，範例：tcp、icmp、udp 及 all，可以在/etc/services 查到所有的 TCP 協定。

範例：
iptables -A INPUT -p icmp --icmp-type ccho-request -s 0/0 -j REJECT

說明：拒絕所有外來的 ping 回應，由於它是使用 ICMP 協定，所以-j REJECT 和 DROP 會丟棄封包，但是會顯示”Port Unreachable”。

使用’-p’來指定協定種類，其中協定分為’TCP’、’UDP’、’ICMP’或是全部（all），

在這的協定寫法沒有分大小寫，並且可以數字代替協定。在/etc/protocols 中有註明各種協定，其中 tcp 為 6、udp 為 17、icmp 為 1。

**TCP 協定**

位於應用層，如果應用程式（http、ftp）需要可靠性高的資料傳輸方式，那麼就可以採用 TCP，它會去檢查資料是否安全到達，否則就重新發送資料。將傳輸的資料以 TCP 格式成為資料段，交由網路層的 IP 協定去處理，每一段資料含有一個檢查值，接收者用它來驗證資料是否受損，如果接收的資料沒有損壞，會傳回確認訊息；如果資料有損壞便會丟棄重發。

用法：

```
-p    tcp    --sport [!]    port [:port]
```

指定 tcp 來源 port，port:port 表示一段範圍的 port 號。

範例：

```
iptables –A OUTPUT –p tcp –s 192.192.73.122 --sport 1024:65535 –j DROPT
```

說明：將 192.192.73.122 連線出去的 port1024-65535 埠號封包通通丟棄。

```
-p    tcp    --dport [!]    port [:port]
```

指定 tcp 目地 port，port:port 表示一段範圍的 port 號。

範例：

```
iptables -A INPUT -d 192.192.73.122 -p tcp --dport 10000 -j ACCEPT
```

說明：將 192.192.73.122 這台主機的 port 10000 開放，允許別人連線進來（port10000 為 webmin 的預設 port）。

```
-p    tcp    --tcp-flags    [!]mask comp
```

指定的 TCP 旗標進行過濾。第一個字串是遮罩（mask）： 檢查的旗標列表。第二個字串是要說哪些東西要設定，旗標有 SYN、ACK、FIN、RST 及 SYN。

```
-p    tcp    [!]—syn
```

為`--tcp-flags SYN,RST,ACK SYN'的簡寫。

範例：

```
iptables -A INPUT -s www.tslg.idv.tw --syn -j DROP
```

說明：拒絕來自 www.tslg.idv.tw 的 TCP 連線，但對連過去不受影響。

**UDP 協定**

位於應用層，讓應用程式直接使用封包傳送服務，如 IP 提供的傳送服務，UDP 協定並不會去檢查封包是否安全到達目的地，且傳送的封包資料量小，因此傳送速度快，但卻是一個不可靠、非連線性的封包協定。

```
-p  udp  --sport [!]  port [:port]
```

指定 tcp 來源 port，port:port 表示一段範圍的 port 號；或是以--sports port,port,prot…表示數個 port 號。

```
-p  udp  --dport [!]  port [:port]
```

指定 tcp 目地 port，port:port 表示一段範圍的 port 號，或是以--dports port,port,prot…表示數個 port 號。

範例：

```
iptables –A INPUT –i eth0 –p 17 –dport 53 –j ACCEPT
```

說明：讓主機提供 DNS 的 port 53 提供連線，-p 17 為 udp 協定。

**ICMP協定**

屬於網路層的一部分，利用 IP 封包的傳送發送它的訊息，ICMP 發送的訊息執行了如偵測遠端機器是否運作（ping）、資料流的控制（當封包到得太快來不及處理時，目的主機傳回一個 ICMP 的來源抑制訊息給發送者，告訴資料來源暫時停止傳送封包）。

雖然 ICMP 沒有 port，但還是有它的選項參數可以使用，並選擇 ICMP 的類型。意即可以指定 ICMP 的名稱或是數字代表(可以執行 iptables –h icmp 來列出詳細的名字）。

一般共同的 ICMP 封包類型為（數字名稱要求的動作）：

- 0 echo-reply Ping
- 3 destination-unreachable 任何的 TCP/UDP 傳輸
- 5 Redirect Routing

- 8 echo-request Ping
- 11 time-exceeded Traceroute

用法（參數簡寫敘述）：
--icmp-type [!] typename

範例：
iptables -A INPUT -p icmp --icmp-type 8 -s www.tslg.idv.tw -j REJECT

說明：拒絕來自 www.tslg.idv.tw 的 ping。

-- jump -j target

當封包符合指定規則時，就執行所指定的 target 目標，如果沒有指定的話，將不會對封包做任何動作。

--in-interface -i [!]name

接收封包網路介面名稱－l0、eth0 或 eth1。

--out-interface -o [!]name

傳送封包出去的網路介面－l0、eth0 或 eth1。

--set-counters -c PKTS BYTES

設定開始的封包數及位元組，用於-A、-I 命令時。

範例：
iptables –A INPUT –c 5 500 ; iptables –L -v

說明：新增一條 INPUT，並將封包數及 bytes 設為 5 及 500。

-m limit --limit rate 指定觸發值

rate 為每幾個單位之幾，單位”1/second”,”1/minute”,”1/hour”,”1/day”可簡寫。

範例：

iptables –A INPUT –p tcp –d 192.192.73.122 --dport 23 –m limit --limit 30/m –j REJECT

說明：讓第一次 telnet 連向 192.192.73.122 時通，再下一次時就拒絕它的連線；次數為通、斷、通…。其中 30/m 為每 60 秒中 30 秒，也就是 1/2 的機會。

-- verbose  -v

完整模式，會列出界面名稱、規則、TOS 偽裝，封包和位元組計數也會列出，須和-L 一起使用。

--numeric  -n

直接使用 IP 顯示，主機名稱、DNS 都會以數字來顯示，一般是建議加上–n，速度會比較快。

--exact  -x

配合-L、-v 選項以精準的去顯示封包、位元組的大小，預設是 k、M、G 為單位，如封包為 5kbytes，加上-x 後為 5000bytes。

--line-numbers

當列出規則時，要在每條規則前加上號碼；如果要刪除規則，直接指定號碼即可。
範例：
iptables -L -n -v --line-numbers

-m state --state 連線的狀態

取得封包連線時的狀態。當封包為無效時表示為”INVAILD”，當封包已經建立連結（telnet 已經連線中）時表示為”ESTABLISHED”，”NEW”表示為一個新的連線（如要 telnet 時），當”RELATED”時表示新的連結及現存的連結，就是說 FTP 的檔案傳輸時或 ICMP（ping），呈現一個斷斷續續的狀態時。

範例：
iptables –A input –i eth0 –m state --state RELATED,ESTABLISHED –j ACCEPT

說明：讓像 ping 動作及連線中的狀態的封包繼續通過。

範例：

```
iptables –A input –i eth0 –m state --state NEW,INVALID –j REJECT
```

說明：拒絕掉外部要進來的連線（新建立）及無效的連線。

```
-m  mac  --mac-source 針對網卡號來指定
```

範例：

```
iptables –A INPUT –m mac –mac-source 00:80:AD:51:04:9E -j DROP
```

說明：拒絕掉來自網卡號碼 00:8d:AD:51:04:9E 機器的封包（如果是已知的話）。

```
-m  tos  --tos tos
```

在 IP 標頭的 8bits Type of Service（服務型態），可以用 iptables -m tos –h 查詢，它提供允許控制資訊具有比一般資料更高優先順序的機制，但大部份主機與路由器都忽略掉。服務類型值為 00（十六進位），表示期望在一般優先權、一般延遲、一般通訊量、一般可靠度及一般成本下進行通訊。

```
-m  ttl  --ttl
```

存活時間（Time to live）用來限制封包可允許經過的中繼站數目，封包經過中繼站時，每個路由器會將 ttl 減 1，直到零。

重要的指定目標

語法：

```
iptables [-t  TABLES]  -ACTION [PATTERN]  [-j  TARGET] [--target 選項]
```

由於 tables 是方式，完全端看要做 filter、nat 及 mangle 哪一種 tables 而已。而 target 主要是對封包做的處理動作，當封包經過規則的比對符合時會對該封包做處理，此動作稱為目標（target）。

iptabless 利用選項'-j'(--jump)去指定目標，若前面設了一長串，卻少了這最後一項，可以說是全功盡棄。在這要注意的是它必須是大寫，分為下列幾項目標：

| 標準的目標 | 目標選項 | 說明 |
| --- | --- | --- |

| | | |
|---|---|---|
| -j ACCEPT | | 讓封包通過 |
| -j DROP | | 丟棄封包 |
| -j QUEUE | | 將封包傳給 userspace 去處理 |
| -j RETURN | | 停止該 chain 並返回 |
| 延伸的目標 target | | |
| -j LOG | --log-level level<br>--log-prefix prefix<br>--log-tcp-sequence<br>-log-tcp-options<br>--log-ip-options | 經由 syslod 記錄封包訊息 |
| -j MARK | --set-mark mark | 用於 mangle table |
| -j REJECT | --reject-with type | 丟棄封包，並傳回錯誤訊息 |
| -j TOS | --set-tos tos | Type Of Service |
| -j MIRROR | | |
| -j SNAT | --to-source ipaddr:port | 轉譯來源位址 |
| -j DNAT | --to-destination ipadr:port | 轉譯目的位址 |
| -j MASQUERADE | --to-ports port | 做 ip 偽裝 |
| -j REDIRECT | --to-ports port | 轉寄封包 |
| -j ULOG | | 給 userspace 記錄封包 |
| -j TCPMSS | | |
| -j TTL | --ttl-set ttl<br>--ttl-del ttl<br>--ttl-inc ttl | 修改 TTL 的值 |

-j ACCEPT

讓封包通過，也就是可以通過規則檢驗而放行、順利通過這個規則鏈。

-j DROP

丟棄封包（DENY）也就不能通過規則檢驗而被擋掉。

-j QUEUE

QUEUE 可以替使用者空間（userspace）行程儲列封包。封包重導至本機端的佇

列（queue）程式。

-j RETURN

RETURN 停止該規則鏈返回或是到下一條規則鏈。

-j LOG

開啟 kernel 來記錄封包，關於 iptables 的訊息會被記錄在/var/log/message 檔案。在標準的 Linux 系統上，kernel 的輸出訊息經由 klogd（kernel logging daemon）所記錄。其中的記錄為：

- --log-level level 記錄的層級，參考 man syslog。
- --log-prefix prefix 訊息前上加上自定的字，最大為 29 字母。
- --log-tcp-squence 記錄 TCP 連續數字。
- --log-tcp-options 記錄 TCP 封包標頭的選項。
- --log-ip-options 記錄 IP 封包標頭的選項。

範例：

```
iptables –A INPUT –j LOG --log-level 6
```

說明：將所有封包進入主機的訊息記錄在/var/log/message 中，且不會列在 console 上。記錄的格式如下圖，其為從 192.168.1.124 的機器使用 ssh 登入主機上的其中一段訊息。當有事發生時，這時所記錄的各種訊息就變的相對重要；不過，由於檔案會越變越大，所以務必要定時清理。

```
Feb 18 14:59:53 net122 kernel: IN=eth1 OUT= MAC=00:48:54:5d:00:4b:00:50:fc:8d:f0:9
a:08:00 SRC=192.168.1.124 DST=192.192.73.122 LEN=52 TOS=0x10 PREC=0x00 TTL=64 ID=1
93 DF PROTO=TCP SPT=49153 DPT=22 WINDOW=33304 RES=0x00 ACK URGP=0
```

-j REJECT

用於 INPUT、FORWARD、OUTPUT。向發送端送出一個“port unreachable”的 ICMP 錯誤訊息（與“DROP”相同）。

--reject-with type

type 為錯誤訊息，預設是”port unreachable”，共有”icmp-net-unreachable”、”icmp-host-unreachable”、”icmp-port-unreachable”、”icmp-proto-unreachable”、”icmp-net-prohibited”、”icmp-host-prohibited”。

範例：

```
iptables –A INPUT –p –icmp –icmp-type 8 –j REJECT –reject-with
icmp-port-unreachable
```

說明：拒絕所有的 ping，並顯示”Destination Port Unreachable”。

每個 type 所對應的訊息如下：
icmp-net-unreachable => Destination Net Unreachable
icmp-host-unreachable => Destination Host Unreachable
icmp-port-unreachable => Destination Port Unreachable
icmp-proto-unreachable => Destination Protocol Unreachable
icmp-net-prohibited => Dest Unreachable, Bad Code: 9
icmp-host-prohibited => Dest Unreachable, Bad Code: 10

-j SNAT
用於 nat tables 中的 POSTROUTING 規則鏈中，然後做內部網路的 NAT，將內部虛擬 ip 轉址出去。

範例：
iptables -t nat -A POSTROUTING –p tcp –s 192.168.1.0/24 –dport 22 –j SNAT --to 192.192.73.122

說明：讓 192.168.1.0/24 網段的機器，能夠藉著真實 ip192.192.73.122 轉址連線出去，到達目的 port22 做 ssh 的連線。

-j DNAT
用於 nat table 中的 PREROUTING 規則鏈中，做封包轉送，不但可以指定轉送到內部虛擬的 ip，也可以轉送到別的 port 上。

範例：
iptables -t nat -A PREROUTING -p tcp -d 192.192.73.122 --dport 22 –j DNAT --to 192.168.1.100:22

說明：將連向 192.192.73.122 的 ssh(port 22)的連線，轉向到內部虛擬主機 192.168.1.100 上的 port 22。

-j MASQUERADE
用於 tablesc 中的 nat 的 POSTROUTING 規則鏈中，告訴 KERNEL 去偽裝封包，用來做 ip 偽裝（NAT）。

範例：

```
iptables –t nat –A POSTROUTING -s 192.168.1.0/24 -j MASQUERADE
```

說明：將來源 192.168.1.X 網段做 ip 偽裝，對外轉址。

-j REDIRECT
用於 nat tables 中的 PREROUTING 和 OUTPUT 規則鏈中，用來轉換 port。

範例：

```
iptables -t nat -A PREROUTING -p tcp –d 192.192.73.122 --dport 80 -j REDIRECT
--to-pors 10000
```

說明：將外部網路連向 192.192.73.122 主機上的 port 80 時，把 port 80 指向 port 10000。

範例
建議將預設的政策通通設為 DROP，然後再一一視需要開放。清除舊有的規則如下：

```
iptables –F
iptables –F –t nat
iptables –F –t mangle
iptables –X
iptables –X –t nat
iptables –X –t mangle
```

- **SNAT（ip masquerade）**：讓虛擬 ip 對外轉址。

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables –t nat –A POSTROUTING –s 192.168.1.0/24 -j MASQUERADE
```

- **開放系統服務**：一般都是使用 tcp，而 DNS 則是使用 udp 協定，”-i eth0”指經由網路卡介面進入的封包，就是指所有的來源–s 0.0.0.0/24。

```
iptables –A INPUT –i eth0 –p tcp --dport 20 –j ACCEPT          // ftp-data
iptables –A INPUT –i eth0 –p tcp --dport 21 –j ACCEPT          // ftp
iptables –A INPUT –i eth0 –p tcp --dport 22 –j ACCEPT          // ssh
iptables –A INPUT –i eth0 –p tcp --dport 23 –j ACCEPT          // telnet
iptables –A INPUT –i eth0 –p tcp --dport 25 –j ACCEPT          //sendmail
iptables –A INPUT –i eth0 –p tcp --dport 53 –j ACCEPT          //DNS
iptables –A INPUT –i eth0 –p udp --dport 53 –j ACCEPT          //DNS
iptables –A INPUT –i eth0 –p tcp --dport 80 –j ACCEPT          //http
iptables –A INPUT –i eth0 –p tcp --dport 110 –j ACCEPT         //pop3
```

```
iptables –A INPUT –i eth0 –p tcp --dport 443 –j ACCEPT          //https
```

- **NAPT**：若不想在同一台主機上同時提供 5、6 種的服務，可以利用 DNAT（NAPT），將系統服務指到其它的主機上。192.192.73.122 的 http 服務，轉到內部的主機 192.168.1.2 的 port 80 上。

```
iptables –t nat –A PREROUTING –p tcp –d 192.192.73.122 --dport 80 –j DNAT --to 192.168.1.2:80
```

- **轉換 port 號**：指定外部電腦經 web 連線到主機的 port80 時，轉向到 port10000（webmin 的預設 port），將外部網路連向 192.192.73.122 主機上的 port 80 時，把 port 80 指向 port 10000。

```
iptables -t nat -A PREROUTING -p tcp –d 192.192.73.122 --dport 80 -j REDIRECT --to-port 10000
```

- **擋掉 ICMP 封包**：
  - 簡易設定

```
iptables –A INPUT –p icmp --icmp-type echo-request –j REJECT
```

  - 進階設定

```
iptables –I INPUT –p imcp --icmp-type echo-request –m limit --limit 6/min --limit-burst 2 –j ACCEPT
```

- **提供 Transparent Proxy（透通式 Proxy）**：
  - 修改 Squid Proxy，使其提供 Transparent Proxy 服務，將以下內容加到 Squid Proxy Server 的設定檔（/etc/squid/squid.conf），完成後再重新啟動 Squid Server。

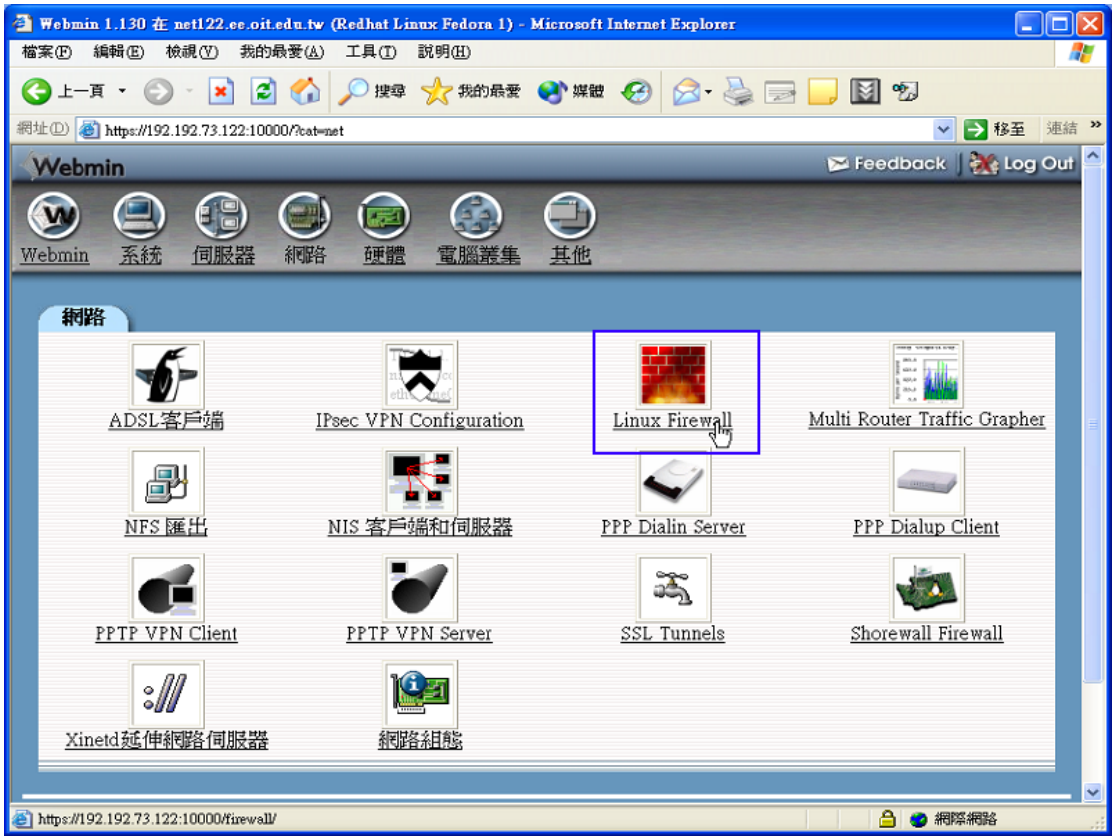
```
http_port 8080
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

  - 設定 iptables 規則

```
iptables –t nat –A PREROUTING –i eth1 –p tcp --dport 80 –j REDIRECT --to-ports 8080
```

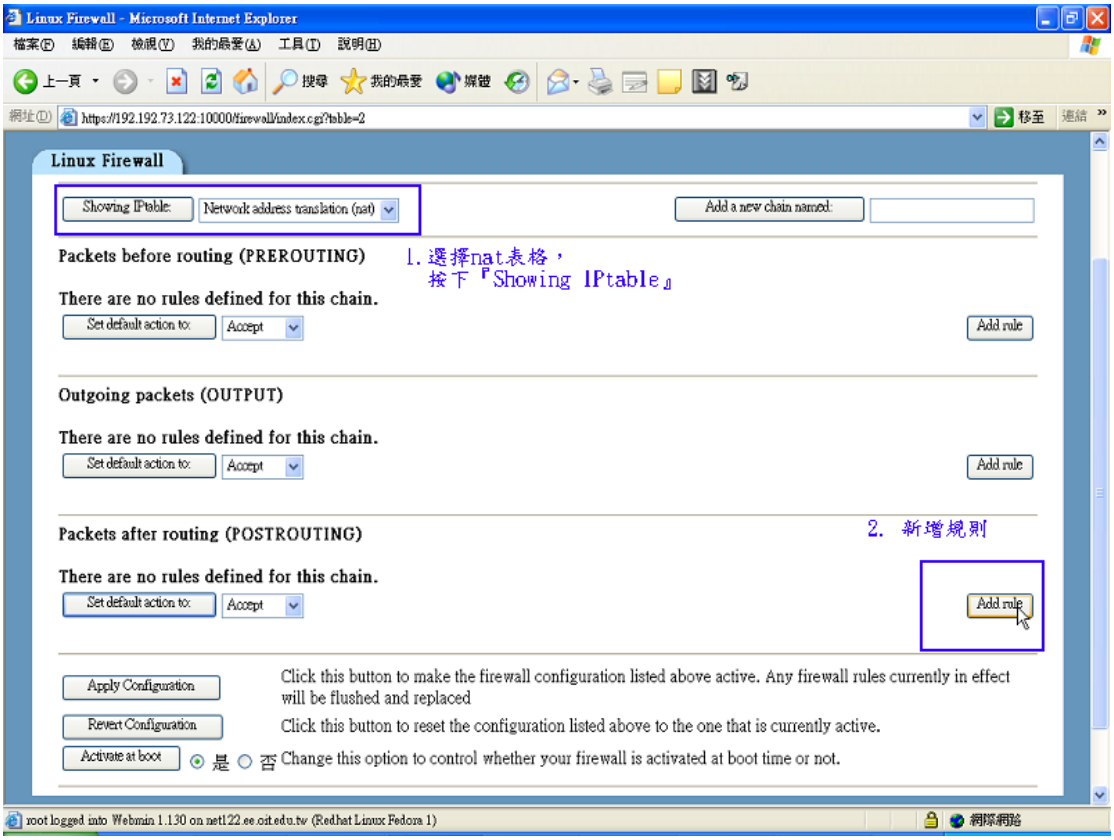
在 Webmin 上使用 iptables

登入 webmin 後，在【網路】選項中看到的【Linux Firewall】選項，即是使用 iptables。



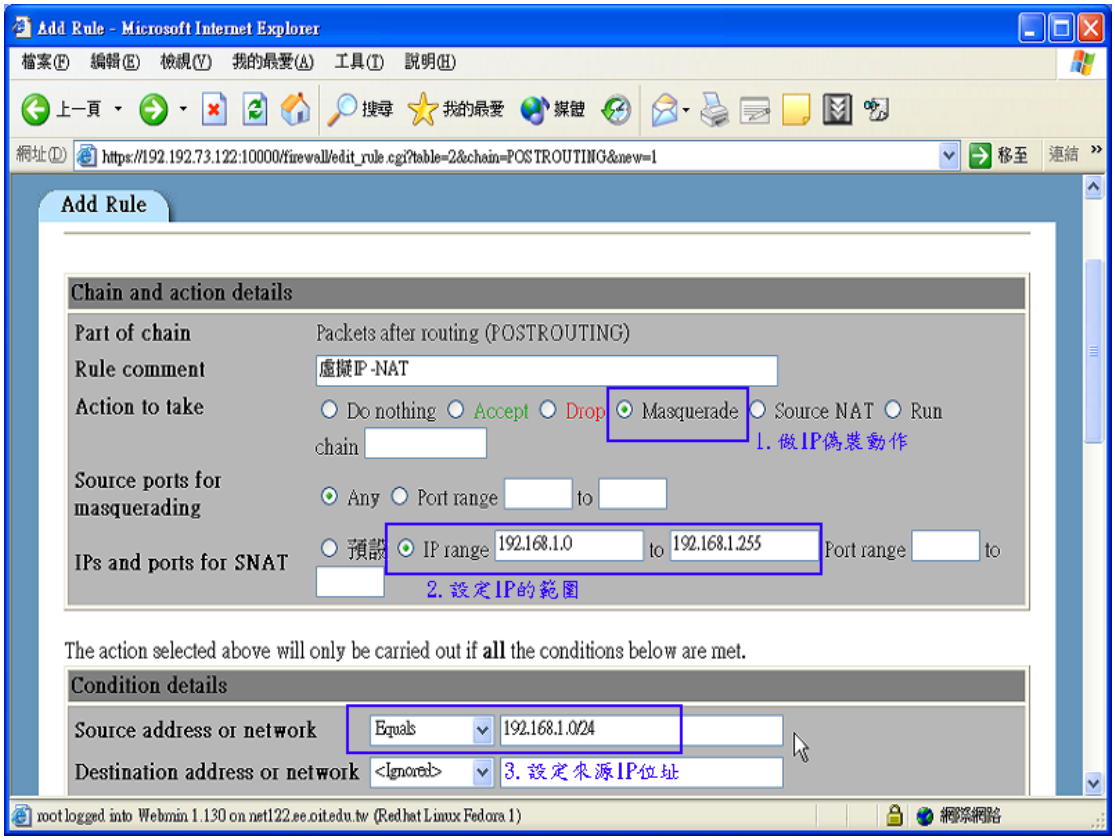

**用 webmin 新增 NAT 規則**

- 範例 1：先選擇 nat table，並按下左邊的『showing iptable』，再按下 POSTROUTING 中的 ”Add rule” 以新增規則。

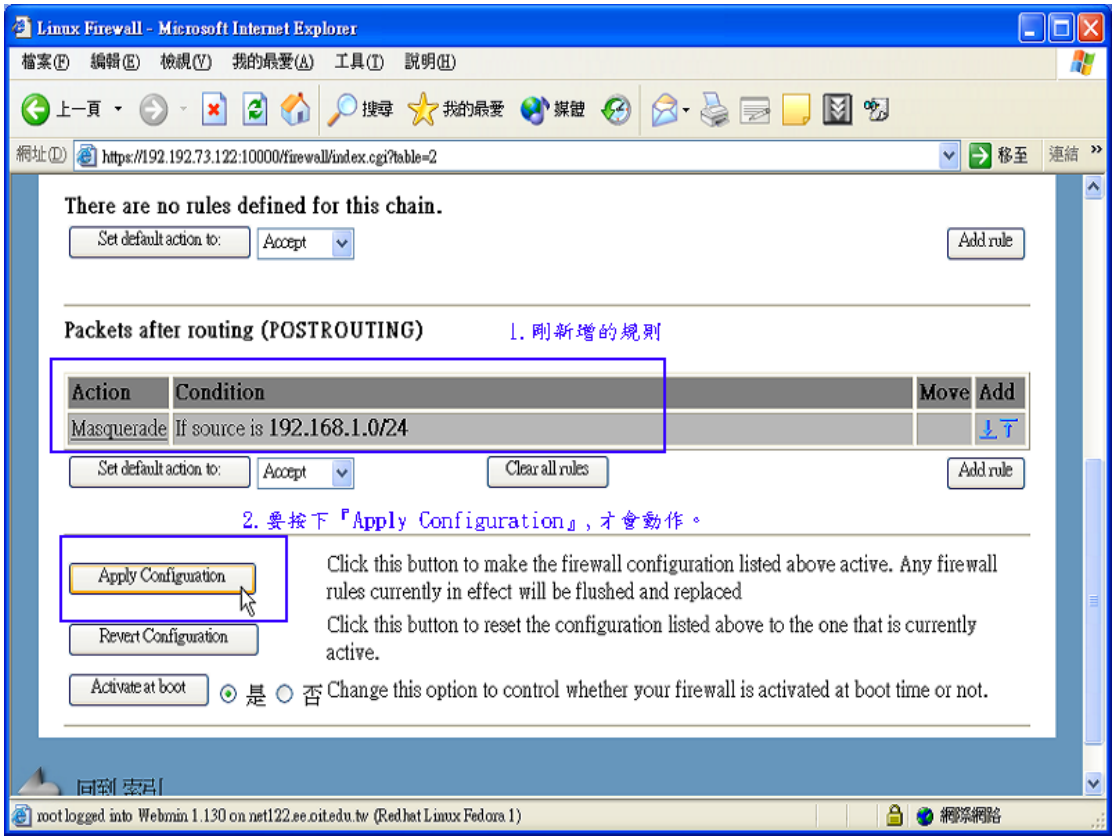



說明：

- 選擇做 IP Masquerade 動作。
- 指定來源的虛擬 IP，port 預設為全部。
- 指定來源虛擬 IP。
- 指定從 eth0 界面出去。

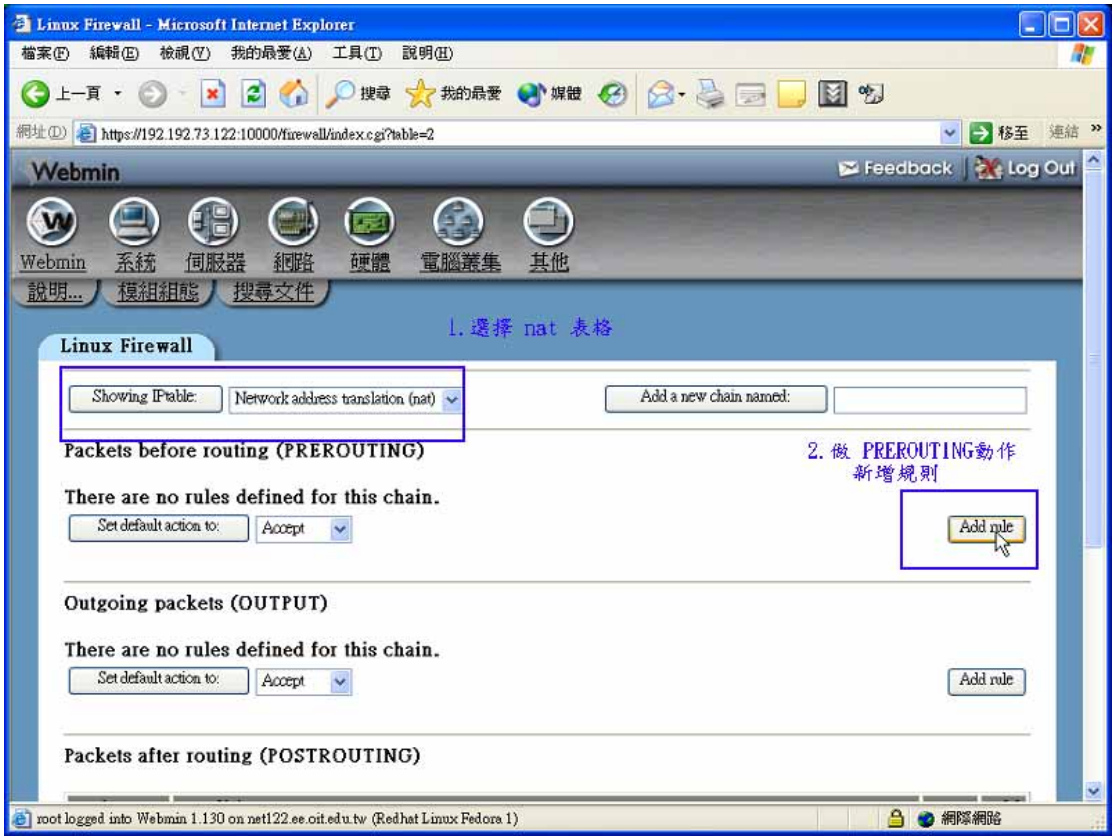



按照以上的方法做完後，將畫面移到最下面，然後按下 建立 建立規則，接著可看到畫面中已出現 IP Masquerade 的規則，接著要去啟動它，按下 Apply Configuration 後表示已經啟動，然後再去修改它。




- 範例二：當外部連到 192.192.73.100 的 HTTP 連線（port 80），轉到內部的 192.168.1.100 的 WEBMIN 連線 port 10000 上.。

選擇 Showing IPtable: Network address translation (nat) ，並按下在「PREROUTING』中的 Add rule 新增規則。

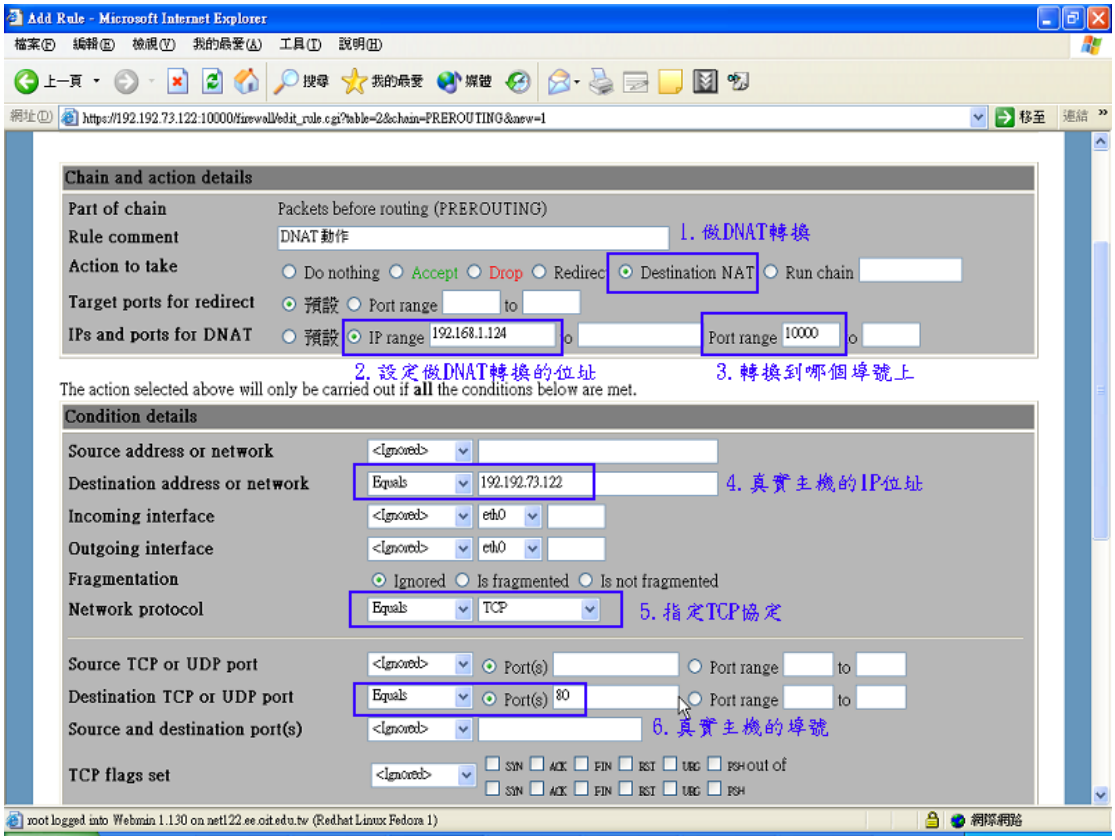



說明：

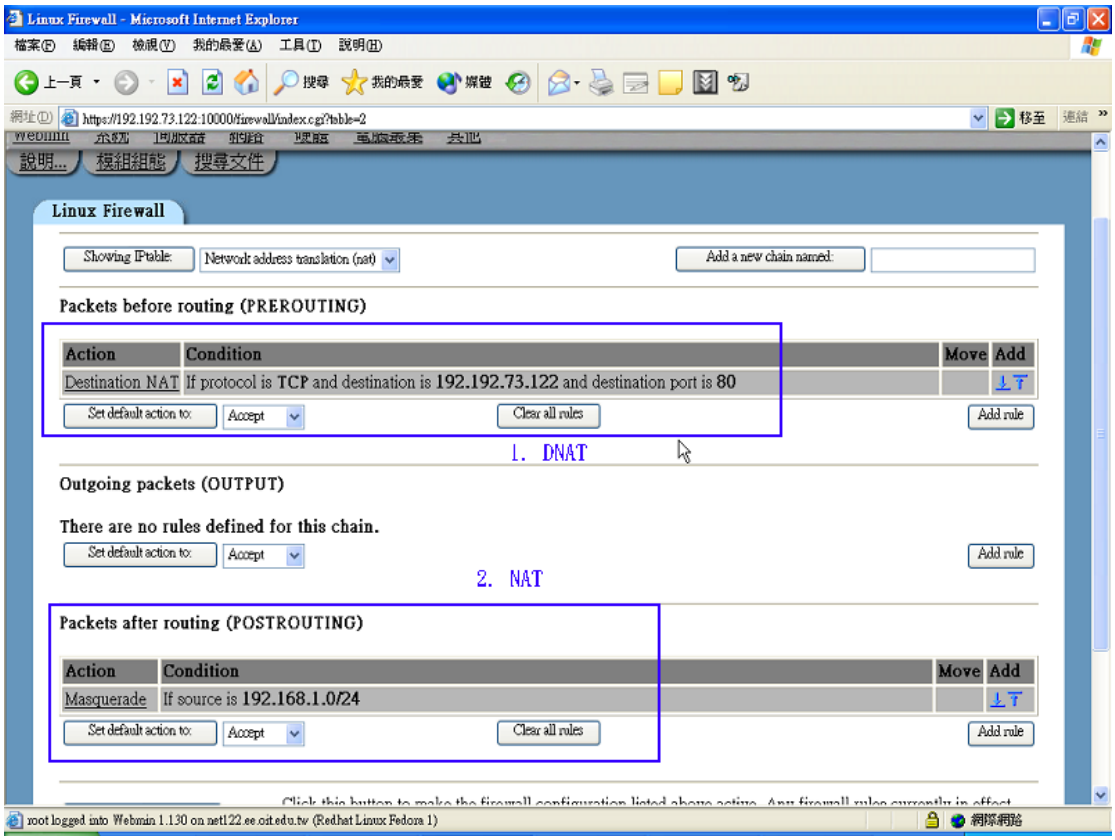
- 選擇做 DNAT 轉換。
- 指定做 DNAT 轉換的內部 IP 及 port。
- 指定真實主機的 IP。
- 指定 tcp 協定。
- 指定真實主機的 port。

其中的運用在於利用這個原則將真實 IP 上的各個 port，對應到各種不同伺服器的主機，以達到分散式的目的。如此一來，當連向 192.192.73.122 這台的 Web 網頁時，就會被轉向到內部 192.168.1.124 這台機器的 Web 網頁上。

之後按下 建立 ，再來是去啟動它，然後按下 Apply Configuration 才表示已經啟動，然後再去修改它。




如下圖所示的畫面是已經建立了兩條 IP Masquerade、DNAT 的規則，如果要建立其它的規則，設定方式完全相同。



## 5.問題與討論

1. 說明 iptables 過濾表的功能。
2. 請利用 vmware 模擬伺服器與 Microsoft 系統來模擬客戶端。該伺服器有三個網路裝置，第一個網路裝置連線對外有一固定 IP 位址，第二、三個網路裝置為私有 IP 位址。該伺服器具有路由器、防火牆功能及 DHCPd，接於第二個網路裝置的網段可透過第一個網路裝置連線對外，接於第三個網路裝置的網段僅可連線至接於第二個網路裝置的網段。
3. 如何利用 iptables 做網路流量管制？
4. 比較利用 iptables 建立的防火牆系統和一般硬體式防火牆的效能。
5. 如何利用 iptables 防止有人亂使用 port scan 軟體（如 nmap）來擾亂自己的 port？