Linux IP Masquerade mini HOWTO 中譯版

作者: Ambrose Au, *ambrose@writeme.com*;
David Ranch, *dranch@trinnet.net*
譯者: 黃志偉 *cwhuang@linux.org.tw*

v1.50, 7 February 1999 翻譯日期: 17-31 March 1999

這份文件描述如何在一台 Linux 主機上起動 IP Masquerade 功能，允許沒有註冊網際網路 IP 位址的連線電腦經由你的 Linux 機器連接網際網路。

目錄

1 簡介

1.1 簡介

這份文件描述如何在一台 Linux 主機上起動 IP Masquerade 功能，允許沒有註冊網際網路 IP 位址的連線電腦經由你的 Linux 機器連接網際網路。你的機器可能是以乙太網路連接 Linux, 也可能是其它種類，像是撥接的點對點(ppp) 連線。這份文件將會強調乙太網路連線的情況，因爲這應該是最常見的案例。

> 這份文件目的是給 **2.2.x** 及 **2.0.x** 穩定版核心的使用者參考。舊版的核心如 **1.2.x** 並不包含在內。

1.2 前言，回饋 & 參考資訊

我發現新手在較新的核心上，像是 2.x 核心，設定 IP Masquerade 時非常困惑。雖然有份常見問答集(FAQ) 與通信論壇(mailing list)，然而沒有一份這方面的專門文件；而且在通信論壇上有些對於這樣一份說明文件(HOWTO) 的請求。所以，我決定撰寫它給所有新手作爲一個起點，並且希望能拋磚引玉，作爲那些非常了解它的使用者建立文件的基礎。如果你認爲我做的不好，不要在意告訴我，這樣我能把它做得更好。

這份文件很多是以原先 Ken Eves 的常見問答集以及 IP Masquerade 通信論壇裡許多有幫助的訊息作爲基礎。特別感謝 Mr. Matthew Driver 在通信論壇中的訊息引發我設立 IP Masquerade 的靈感以及最後撰寫了這份文件。

如果我的任何資訊有誤或遺漏任何資訊，請別介意把任何回饋或意見寄到 *ambrose@writeme.com* 及 *dranch@trinnet.net* 來。你的無價回饋將影響這份說明文件的未來!

這份說明文件是想作爲讓你的 **IP Masquerade** 能在最短時間內運作的快速指引。 因爲我不是一位專門作家，你可能會發現本文件的資訊並非如你想的那麼一般及客觀。 最新的消息以及資訊可以在我所

維護的 ***IP Masquerade Resource*** <http://ipmasq.cjb.net/> 網頁上找到。 如果你有任何關於 **IP Masquerade** 的技術問題，請加入 **IP Masquerade** 通信論壇而別寄電子郵件給我，因爲我的時間有限，而且 **IP Masquerade** 的發展者們更有能力回答你的問題。

這份文件最新的版本可以在 *IP Masquerade Resource* 上找到，裡面也有 HTML 以及 postscript 的版本:

- *http://ipmasq.cjb.net/*
- *http://ipmasq2.cjb.net/*
- 請參考 *IP Masquerade Resource* 映射站台列表 <http://ipmasq.cjb.net/index.html##mirror> 以找到其它的映射站台。

1.3 版權 & 宣告

這份文件版權屬於 `Ambrose Au 1999`, 而且是免費的文件。你可以在 GNU 的通用公開授權方式下散播它。

這份文件中的資訊跟其它內容都已經盡了我最大的努力。無論如何，IP Masquerade 是實驗性的，而且我也可能會犯些錯誤；所以你應該自己決定是不是要照著這份文件中的資訊做。

沒有人會爲使用這份文件中的資訊所造成的電腦損壞或其它損失負責。也就是說，

> 作者及維護者不對依照這份文件內容動作所造成的損害負責。

原文

This document is `copyright(c) 1996 Ambrose Au`, and it's a free document. You can redistribute it under the terms of the GNU General Public License.

The information and other contents in this document are to the best of my knowledge. However, ip_masq is *experimental*, and there is chance that I make mistakes as well; so you should determine if you want to follow the information in this document.

Nobody is responsible for any damage on your computers and any other losses by using the information on this document. i.e.

> **THE AUTHOR AND MAINTAINERS ARE NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THE INFORMATION IN THIS DOCUMENT.**

2 背景知識

2.1 什麼是 IP Masquerade?

IP Masquerade 是 Linux 的一項網路功能。如果一台 Linux 主機使用 IP Masquerade 功能連線到網際網路上，那麼接上它的電腦（不論是在同一個區域網路上或藉由數據機連線）也可以接觸網際網路，即使它們沒有獲得正式指定的 *IP* 位址。

這使得一些電腦可以隱藏在閘道(gateway) 系統後面存取網際網路而不被發現，看起來就像只有這個系統在使用網際網路。突破設定良好的僞裝(masquerade)系統之安全防護應該會比突破良好的封包過濾式防火牆(packet filter firewall)來得更加困難（假設兩者之中都沒有錯誤）。

2.2 現況

IP Masquerade 已經發展多年而趨於成熟，因此已正式併入 Linux 2.2.x 的核心中。 從 1.3.x 版的核心開始已經內建這項支援。許多個人甚至公司正在使用它，而有滿意的結果。

覽網頁以及遠端簽入(telnet)已經有回報表示可以在 IP Masquerade 上運作。檔案傳輸(FTP)，網路交談(IRC) 以及聆聽 Real Audio 現在可以載入某些模組配合。其它的網路資料流音訊 (streaming audio)像是 True Speech 以及 Internet Wave 也能運作。一些通信論壇中的使用夥伴甚至還嘗試過視訊會議軟體。 Ping 現在配合新近可以取得的網際網路控制訊息協定(ICMP)修補檔也能運作。

更完整的支援軟體列表請參考 4.3 節。

IP Masquerade 在數種不同的作業系統及平臺上與 '客户端機器' 配合良好。 成功的案例有使用 Unix, Windows95, Windows NT, Windows for Workgroup (with TCP/IP package), OS/2, Macintosh System's OS with Mac TCP, Mac Open Transport, DOS with NCSA Telnet package, VAX, Alpha with Linux, 甚至 Amiga with AmiTCP 或 AS225-stack 的系統。 這列表還在不斷增加當中。重點就是，只要你的系統使用 TCP/IP 協定，它就應該能和 IP Masquerade 一起工作。

2.3 誰可以從 IP Masquerade 中獲益?

- 如果你有台連接網際網路的 Linux 主機，而且
- 如果你有一些執行 TCP/IP 連接到 Linux 機器的電腦在區域網路上，以及/或是
- 如果你的 Linux 主機有一個以上的數據機並且作爲 PPP 或 SLIP 伺服器連接其它電腦，它們
- 這些其它機器沒有正式指定的 IP 位址。（這些機器從這裡開始以後就稱爲其它機器）
- 而且當然，如果你希望這些其它機器不必花額外的費用就能連上網際網路 :)

2.4 誰不需要 IP Masquerade?

- 如果你的機器是單獨一台(stand-alone) 連接網際網路的 Linux 主機，那麼執行 IP Masquerade 沒什麼意義，或者
- 如果你的其它機器擁有正式指定的 IP 位址，那麼你就不需要 IP Masquerade
- 而且當然，如果你不喜歡免費使用(free ride) 這個主意的話。

2.5 IP Masquerade 是如何運作的?

節自 Ken eves 的 IP Masquerade FAQ:

這是大部分簡單的設定草圖:

```
 SLIP/PPP         +------------+                      +-------------+
 to provider      |  Linux     |       SLIP/PPP       | Anybox      |
<---------- modem1|            |modem2 ----------- modem |             |
  111.222.333.444 |            |          192.168.1.100 |             |
                  +------------+                      +-------------+
```

上面的草圖中一台安裝並執行 ip_masquerading 的 Linux 機器使用 modem1 經由 SLIP/or/PPP 連接網際網路。它有一個指定的 IP 位址 111.222.333.444。它設定 modem2 允許撥接者簽入並起始 SLIP/or/PPP 連結。

第二個系統（不必是執行 Linux 的系統）撥接進入 Linux 機器並起始 SLIP/or/PPP 連結。它在網際網路上並沒有指定的 IP 位址所以它使用 192.168.1.100。（參閱下述）

配合 ip_masquerade 及適當遞送配置(routing configured) Anybox 這台機器可以跟網際網路交流就如同它眞的連在上面般（除了少數例外）。

節錄 Pauline Middelink:
別忘記提到 ANYBOX 應該把 Linux 機器當作它的閘道（無論是預設遞送路徑或只是個子網路都沒關係）。如果 ANYBOX 不能夠這樣設， Linux 機器應該爲所有要遞送的位址做代理位址解析析協定(proxy arp) 服務，但代理位址解析的設定超過這份文件的範圍。

下面節錄自 comp.os.linux.networking 的一篇布告並且稍加編輯以符合上述範例的用詞:

- 我告訴 ANYBOX 這台機器跑 slip 的 linux 機器是它的閘道。
- 當一個封包從 ANYBOX 進入 linux 機器時，它會指定新的來源埠號(source port number)，把它自己的 ip 位址塞入封包的標頭並儲存原來的。然後它將會藉由 SLIP/or/PPP 界面把修改過的封包送上網際網路。
- 當一個封包從網際網路來到 linux 機器時，如果埠號是上面指定的其中一個，它將會取出原來的埠號以及 ip 位址，把它們放回封包的標頭，並且把封包送往 ANYBOX。
- 送出封包的主機將永遠不知道其中的差別。

一個 IP Masquerading 的例子:

下面的圖示是典型的例子:-

```
+----------+
|          |  Ethernet
| abox     |::::::
|          |2    :192.168.1.x
+----------+     :
                 :   +----------+   PPP
+----------+     :  1|  Linux   |   link
|          |     ::::| masq-gate|:::::::::// Internet
| bbox     |::::::   |          |
```

```
|           |3     :   +----------+
+----------+      :
                  :
+----------+      :
|          |      :
| cbox     |::::::
|          |4
+----------+


<-Internal Network->
```

在這個例子中我們考慮四台電腦系統（想必遙遠的右方還有些東西讓你到網際網路的 IP 連線能夠連接，以及一些（遠超過這一頁）在網際網路上你有興趣交換資訊的東西）。 這個 Linux 系統 `masq-gate` 是 `abox`, `bbox`, `cbox` 內部網路機器連接網際網路的偽裝閘道。 內部網路使用指定的私用(private) 網路位址，在這個案例中是 class C 網路 192.168.1.0, Linux 機器擁有位址 192.168.1.1 而其它系統也擁有此網路上的位址。

這三台機器 `abox`, `bbox` 以及 `cbox` (它們可以執行任何作業系統 — 像是 **Windows 95**, **Macintosh MacTCP** 或甚至是另一台 Linux 機器，只要它們能了解 IP)可以連線到網際網路上的其它機器去，然而這個偽裝系統閘道 `masq-gate` 轉換它們所有的連線所以這些連線看起來像是原本即從偽裝閘道 `masq-gate` 本身發出的，而且還安排偽裝連線傳回的資料轉回原先的系統 — 所以在內部網路上的系統看到的是直接通往網際網路的遞送路徑而且不知道他們的資料被偽裝過。

2.6 在 Linux 2.2.x 上使用 IP Masquerade 的需求

** 請參考 ***IP Masquerade Resource*** <http://ipmasq.cjb.net/> 以獲得最新資訊。**

- 2.2.x 版核心的原始程式碼可從 *http://www.kernel.org/* 取得。
  (大部份現代的發行套件如 Red Hat 5.2 - 配備了 2.0.36 的核心 - 已有了模組化的核心，且所有 IP Masquerade 所需的選項都已打開。 這種情形下，已沒有必要再自行編譯。如果你自己升級核心，你應注意你需要什麼，在文件稍後會提到。)
- 可載入的核心模組，最好是 2.1.121 或以後的版本
- 設定良好的 TCP/IP 網路
  相關文件在 *Linux NET-3 HOWTO* <http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html> 及 *Network Administrator's Guide* <http://metalab.unc.edu/mdw/LDP/nag/nag.html>
  也請參考 *Trinity OS Doc* <http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS.wri>, 是一份非常完整的 Linux 網路參考指引。
- 將你的 Linux 主機連上網際網路
  相關文件有 *Linux ISP Hookup HOWTO* <http://metalab.unc.edu/mdw/HOWTO/ISP-Hookup-HOWTO.html>, *Linux PPP HOWTO* <http://metalab.unc.edu/mdw/HOWTO/PPP-HOWTO.html>, *Linux DHCP mini-HOWTO* <http://metalab.unc.edu/mdw/HOWTO/mini/DHCP.html> 及 *Linux Cable Modem mini-HOWTO* <http://metalab.unc.edu/mdw/HOWTO/mini/Cable-Modem.html>

- IP Chains 1.3.8 或更新的版本。可從 *http://www.rustcorp.com/linux/ipchains/* 取得。
  版本需求的更多資訊也在 *Linux IP Firewalling Chains page* <http://www.rustcorp.com/linux/ipchains/>。
- 其它的資訊，請見 *Linux IP Masquerade Resource* <http://ipmasq.cjb.net/>

2.7 在 Linux 2.0.x 上使用 IP Masquerade 的需求

** 請參考 ***IP Masquerade Resource*** <http://ipmasq.cjb.net/> 以獲得最新資訊。**

- 核心 2.0.x 的原始程式碼可以從這裡取得 *http://www.kernel.org/*
  (大部份現代的發行套件如 Red Hat 5.2 - 配備了 2.0.36 的核心 - 已有了模組化的核心，且所有 IP Masquerade 所需的選項都已打開。 這種情形下，已沒有必要再自行編譯。如果你自己升級核心，你應注意你需要什麼，在文件稍後會提到。)
- 可載入核心模組，最好是 2.0.0 或更新的版本，可以從這裡取得 *http://www.pi.se/blox/modules/modules-2.0.0.tar.gz*
  (至少需要 modules-1.3.57)
- 設定好的 TCP/IP 網路

  相關文件在 *Linux NET-3 HOWTO* <http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html> 及 *Network Administrator's Guide* <http://metalab.unc.edu/mdw/LDP/nag/nag.html>
  也請參考 *Trinity OS Doc* <http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS.wri>, 是一份非常完整的 Linux 網路參考指引。
- 將你的 Linux 主機連上網際網路
  相關文件 *Linux ISP Hookup HOWTO* <http://metalab.unc.edu/mdw/HOWTO/ISP-Hookup-HOWTO.html>, *Linux PPP HOWTO* <http://metalab.unc.edu/mdw/HOWTO/PPP-HOWTO.html>, *Linux DHCP mini-HOWTO* <http://metalab.unc.edu/mdw/HOWTO/mini/DHCP.html> 及 *Linux Cable Modem mini-HOWTO* <http://metalab.unc.edu/mdw/HOWTO/mini/Cable-Modem.html>
- Ipfwadm 2.3 或更新的版本可以從這裡取得
  ftp://ftp.xos.nl/pub/linux/ipfwadm/ipfwadm-2.3.tar.gz 在 Linux Ipfwadm 網頁上有更多關於版本的資訊 *Linux IPFWADM page* <http://www.xos.nl/linux/ipfwadm/>
- 你可以選擇性地加上一些 IP Masquerade 修補檔以增加其它功能。 從這裡可以取的更多資訊，*IP Masquerade Resources* <http://ipmasq.cjb.net/> (這些修補檔適用於所有的 2.0.x 核心)

3 IP Masquerade 的設定

如果你的私用網路裡有任何重要的資訊，在使用 **IP Masquerade** 之前請三思。這可能成為你通往網際網路的閘道，反之亦然，也可能成為另一邊的世界進入你私用網路的途徑。

3.1 編譯核心加入 IP Masquerade 的支援

如果你的 **Linux** 發行套件已經將下面將提到的所需特性及模組編譯進去的話(大部份模組化的核心有你所需的東西)，那麼你不需要重新編譯核心。 不過仍十分建議你讀一讀此節，因為它包含了其它有用的資訊。

3.1.1 Linux 2.2.x 版核心

- 首先，你需要 2.2.x 版核心的原始程式碼。
- 如果這是你第一次編譯核心，不要害怕。 事實上，這非常容易而且涵蓋於 *Linux Kernel HOWTO* 。
- 以這個指令: `tar xvzf linux-2.2.x.tar.gz -C /usr/src` 將核心原始碼解開至 `/usr/src/`, 其中 x 是 2.2 之後的修補層級(確定有一叫 `linux` 的目錄或符號連結)。
- 加上適當的修補。因爲新的修補檔不斷出來，所以細節不會包含在這裡。 最新的資訊請參考 *IP Masquerade Resources* `<http://ipmasq.cjb.net/>`。
- 有關編譯核心更進一步的介紹請參考 Kernel HOWTO 以及核心原始程式碼目錄裡的 README 檔案。
- 這裡是你要編譯進去的選項: 下列選項要回答 *YES*:

```
* Prompt for development and/or incomplete code/drivers
  CONFIG_EXPERIMENTAL
  - 這將讓你能選擇把實驗性的 IP Masquerade 程式碼編譯到核心裡去

* Enable loadable module support
  CONFIG_MODULES
  - 讓你能夠載入 ipmasq 相關模組如 ip_masq_ftp.o

* Networking support
  CONFIG_NET

* Network firewalls
  CONFIG_FIREWALL

* TCP/IP networking
  CONFIG_INET

* IP: forwarding/gatewaying
  CONFIG_IP_FORWARD

* IP: firewalling
  CONFIG_IP_FIREWALL

* IP: masquerading
  CONFIG_IP_MASQUERADE

* IP: ipportfw masq support
  CONFIG_IP_MASQUERADE_IPPORTFW
  - 建議加入

* IP: ipautofw masquerade support
  CONFIG_IP_MASQUERADE_IPAUTOFW
  - 可選用

* IP: ICMP masquerading
```

```
CONFIG_IP_MASQUERADE_ICMP
- 支援偽裝 ICMP 封包，建議加入

* IP: always defragment
  CONFIG_IP_ALWAYS_DEFRAG
  - 高度建議使用

* Dummy net driver support
  CONFIG_DUMMY
  - 建議加入

* IP: ip fwmark masq-forwarding support
  CONFIG_IP_MASQUERADE_MFW
  - 可選用
```

注意: 這只不過是你跑 IP Masquerade 所需的項目，其它的按照你所需的來選擇。

- 在編譯完核心之後，你需要編譯及安裝模組:

```
make modules; make modules_install
```

- 然後把下面幾行加到你的 /etc/rc.d/rc.local 檔案 (或其它你認爲適當的檔案)中以便於每次重開機時自動載入位於 /lib/modules/2.2.x/ipv4/ 的模組:

```
        .
        .
        .
/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
(以及其它的模組如 ip_masq_cuseeme, ip_masq_vdolive,
如果你有加上這些修補的話)
        .
        .
        .
```

重要: **IP** 轉送功能在 **2.2.x** 版核心中預設是關閉的，請確定你以此指令將其打開:

```
echo "1" > /proc/sys/net/ipv4/ip_forwarding
```

對於 Red Hat 的使用者來說，你可以把 /etc/sysconfig/network 中的 FORWARD_IPV4=false 改成 FORWARD_IPV4=true。

- 重新啓動 Linux 主機。

3.1.2 Linux 2.0.x Kernels

- 首先，你需要核心的原始程式碼(最好是最新的 2.0.36 或以上的版本)。
- 如果這是你第一次編譯核心，不要害怕。 事實上，這非常容易而且涵蓋於 *Linux Kernel HOWTO* 。

- 以這個指令: `tar xvzf linux-2.0.x.tar.gz -C /usr/src` 將核心原始碼解開至 `/usr/src/`, 其中 x 是 2.0 之後的修補層級(確定有一叫 `linux` 的目錄或符號連結)。
- 加上適當的修補。因爲新的修補檔不斷出來，所以細節不會包含在這裡。 最新的資訊請參考 *IP Masquerade Resources* `<http://ipmasq.cjb.net/>`。
- 有關編譯核心更進一步的介紹請參考 Kernel HOWTO 以及核心原始程式碼目錄裡的 README 檔案。
- 這裡是你要編譯進去的選項: 下列選項要回答 *YES*:

```
* Prompt for development and/or incomplete code/drivers
  CONFIG_EXPERIMENTAL
  - 這將讓你能選擇把實驗性的 IP Masquerade 程式碼編譯到核心裡去

* Enable loadable module support
  CONFIG_MODULES
  - 讓你能夠載入模組

* Networking support
  CONFIG_NET

* Network firewalls
  CONFIG_FIREWALL

* TCP/IP networking
  CONFIG_INET

* IP: forwarding/gatewaying
  CONFIG_IP_FORWARD

* IP: firewalling
  CONFIG_IP_FIREWALL

* IP: masquerading (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE
  - 這雖然是實驗性的，但卻是 *必須* 的

* IP: ipautofw masquerade support (EXPERIMENTAL)
  CONFIG_IP_MASQUERADE_IPAUTOFW
  - 建議加入

* IP: ICMP masquerading
  CONFIG_IP_MASQUERADE_ICMP
  - 支援僞裝 ICMP 封包，可選用

* IP: always defragment
  CONFIG_IP_ALWAYS_DEFRAG
  - 高度建議使用

* Dummy net driver support
  CONFIG_DUMMY
```

- 建議加入

注意: 這只不過是你跑 IP Masquerade 所需的項目，其它的按照你所需的來選擇。

- 在編譯完核心之後，你需要編譯及安裝模組:

```
make modules; make modules_install
```

- 然後把下面幾行加到你的 /etc/rc.d/rc.local 檔案 (或其它你認爲適當的檔案)中以便於每次重開機時自動載入位於 /lib/modules/2.0.x/ipv4/ 的模組:

```
        .
        .
        .
/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
(以及其它的模組如 ip_masq_cuseeme, ip_masq_vdolive,
如果你有加上這些修補的話)
        .
        .
        .
```

重要: IP 轉送功能在 2.0.34 之後的核心中預設是關閉的，請確定你以此指令將其打開:

```
echo "1" > /proc/sys/net/ipv4/ip_forwarding
```

對於 Red Hat 的使用者來說，你可以把 /etc/sysconfig/network 中的 FORWARD_IPV4=false 改成 FORWARD_IPV4=true。

- 重新啓動 Linux 主機。

3.2 指定私用網路的 IP 位址

因爲所有其它機器都沒有正式指定的位址，必須有個正確的方式來分配位址給這些機器。

節自 IP Masquerade FAQ:

有份 RFC (#1597, 現在可能已過時了) 是有關沒有與外界連線的網路該使用什麼 IP 位址。有三個數字區塊是特別爲這個目的而保留的。其中一個我使用的是 192.168.1.n 到 192.168.255.n 之間的 255 Class-C 子網路。

```
節自 RFC 1597:

第三節: 私用位址空間

    網際網路位址指定當局(IANA: Internet Assigned Numbers Authority)
    已經保留下列三個區塊的 IP 位址空間給私用網路:

                    10.0.0.0        -   10.255.255.255
                    172.16.0.0      -   172.31.255.255
```

```
              192.168.0.0     -   192.168.255.255

        我們將稱第一個區塊爲 "24位元區塊"，第二個爲 "20位元區塊"，
        而第三個則稱爲 "16位元區塊"。注意到第一個區塊就只是個
        class A  網路號碼，第二個區塊則是連續的 16 個 class B 網路
        號碼，而第三個區塊是一組 255  個連續的 class C 網路號碼。
```

所以，如果你要使用一個 class C 網路的話，那麼你的機器應該以 192.168.1.1, 192.168.1.2, 192.168.1.3, ..., 192.168.1.x 來名之。

192.168.1.1 通常是閘道這台機器，在此即你連上網際網路的 Linux 主機。注意 192.168.1.0 以及 192.168.1.255 分別爲網路以及廣播位址，是保留的。避免在你的機器上使用這些位址。

3.3 配置其它機器

除了爲每台機器設定適當的 IP 位址之外，你也應該設定適當的閘道。一般說來，這是非常直接了當的。你只需簡單地輸入 Linux 主機的位址(通常是 192.168.1.1)作爲閘道位址。

關於領域名稱服務，你可以加入任何 DNS 系統。最可能的應該是你 Linux 使用的那一個。你也可以選擇性地加上任何網域字尾(domain suffix)。

在你重新配置這些 IP 位址之後，記得重新啓動適當的服務或是重新開機。

下面的配置範例假設你使用一個 Class C 網路並且以 192.168.1.1 作爲 Linux 主機的位址。請注意 192.168.1.0 及 192.168.1.255 是保留的。

3.3.1 配置 Windows 95

1. 如果你還沒有安裝網路卡以及界面驅動程式，現在做。
2. 到 '控制台/網路' 裡去。
3. 如果你的網路配置裡沒有 *'TCP/IP* 協定' 則加進去。
4. 在 *'TCP/IP* 內容'中，選擇 *'IP* 位址'並且把 IP 位址設定爲 192.168.1.x,(1<x<255)，並且把子網路遮罩設爲 255.255.255.0
5. 在 '通訊閘'中加入 192.168.1.x 作爲你的閘道。
6. 在 *'DNS* 配置'/*'DNS* 伺服器'下加入你的 Linux 主機使用的 DNS (通常可以在 /etc/resolv.conf 裡找到)。你可以選擇性地加入適當的網域字尾搜尋順序。
7. 不要變更原先的其它設定，除非你知道自己在做什麼。
8. 在所有的對話盒中按下 '確定'並且重新啓動系統。
9. 測試網路連線，`Ping` 你的 linux 主機: 從 '開始/執行'，輸入 ping 192.168.1.1
(這只是區域網路連線測試，你現在還不能 `ping` 外面的世界。)
10. 你可以在 windows 目錄下選擇性地建立一個 `HOSTS` 檔案，如此你可以使用區域網路裡的機器名稱。在 windows 目錄裡有個稱爲 `HOSTS.SAM` 的範例。

3.3.2 配置 Windos for Workgroup 3.11

1. 如果你還沒有安裝網路卡以及界面驅動程式，現在做。
2. 如果你還未安裝 TCP/IP 32b 套件的話就裝吧。
3. 在 *'Main'/'Windows Setup'/'Network Setup'*, 按下 *'Drivers'*。
4. 將 *'Network Drivers'* 裡的 *'Microsoft TCP/IP-32 3.11b'* 反白，按下 *'Setup'*。
5. 設定 IP 位址於 192.168.1.x (1 < x < 255), 然後設定 Subnet Mask 爲 255.255.255.0 以及 Default Gateway 爲 192.168.1.1。
6. 不要開啓 *'Automatic DHCP Configuration'* 並在 *'WINS Server'* 中放入任何東西，除非你在一 Windows NT 網域中而且你知道你在做什麼。
7. 按下 *'DNS'*, 填入在 3.3.1 小節中步驟六提到的資訊，然後在你完成後按下 *'OK'* 鈕。
8. 按下 *'Advanced'*, 如果你使用類似 3.3.1 小節步驟十中提到主機檔案，勾選 *'Enable DNS for Windows Name Resolution'* 及 *'Enable LMHOSTS lookup'*。
9. 在所有對話盒中按 *'OK'* 並重新啓動系統。
10. `Ping` 一下你的 Linux 主機以測試網路連接: 在 *'File/Run'* 輸入: `ping 192.168.1.1`
    (這只不過是區域網路的連接測試，你還不能 `ping` 到外面的世界)。

3.3.3 Configuring Windows NT

1. 如果你還沒有安裝網路卡以及界面驅動程式，現在做。
2. 到 *'Main'/'Control Panel'/'Network'*。
3. 如果你還沒裝 TCP/IP 服務的話從 *'Add Software'* 選單中加入 TCP/IP 協定及相關的部份。
4. 在 *'Network Software and Adapter Cards'* 裡將 *'Installed Network Software'* 選擇盒中的 *'TCP/IP 協定'* 反白。
5. 在 *'TCP/IP Configuration'*，選擇適當的界面驅動程式，例如，`[1]Novell NE2000 Adapter`。然後設定 IP 位址於 192.168.1.x (1 < x < 255)，然後設定 Subnet Mask 爲 255.255.255.0 以及 Default Gateway 爲 192.168.1.1。
6. 不要開啓 *'Automatic DHCP Configuration'* 並在 *'WINS Server'* 中放入任何東西，除非你在一 Windows NT 網域中而且你知道你在做什麼。
7. 按下 *'DNS'*, 填入在 3.3.1 小節中步驟六提到的資訊，然後在你完成後按下 *'OK'* 鈕。
8. 按下 *'Advanced'*, 如果你使用類似 3.3.1 小節步驟十中提到主機檔案，勾選 *'Enable DNS for Windows Name Resolution'* 及 *'Enable LMHOSTS lookup'*。
9. 在所有對話盒中按 *'OK'* 並重新啓動系統。
10. `Ping` 一下你的 Linux 主機以測試網路連接: 在 *'File/Run'* 輸入: `ping 192.168.1.1`
    (這只不過是區域網路的連接測試，你還不能 `ping` 到外面的世界)。

3.3.4 配置 UNIX 系列的系統

1. 如果你還未安裝你的網路卡並以適當的界面驅動程式重新編譯你的核心，現在就做吧。
2. 安裝 TCP/IP 網路，像是 nettools 套件，如果你還沒裝的話。
3. 將 *IPADDR* 設爲 192.168.1.x ($1 < x < 255$), 然後將 *NETMASK* 設爲 255.255.255.0, *GATEWAY* 設爲 192.168.1.1, 以及 *BROADCAST* 設爲 192.168.1.255。
   例如，在 Red Hat Linux 系統上你可以編輯 `/etc/sysconfig/network-scripts/ifcfg-eth0`，或直接從 Control Panel 裡做。
   (在 SunOS, BSDi, Slackware Linux, 等中都不相同...)
4. 將你的名稱伺服器及領域搜尋字尾加到 `/etc/resolv.conf`。
5. 依據你的設定你可能要更新你的 `/etc/networks` 檔案。
6. 重新啓動適當的服務，或簡單的重新開機。
7. 發出 `ping` 指令: `ping 192.168.1.1` 以測試到你的 gateway 機器的連接性。
   (這只不過是區域網路的連接測試，你還不能 `ping` 到外面的世界)。

3.3.5 配置使用 NCSA Telnet 套件的 DOS 機器

1. 如果你還沒有安裝網路卡，現在做。
2. 載入適當的封包驅動程式。對於 NE2000 卡來說，如果你的卡設定爲 IRQ 10 及硬體位址於 0x300，用 `nwpd 0x60 10 0x300`。
3. 建立一新目錄，然後解開 NCSA Telnet 套件: `pkunzip tel2308b.zip`
4. 使用文字編輯器打開 `config.tel` 檔案。
5. 設定 `myip=192.168.1.x` ($1 < x < 255$), 以及 `netmask=255.255.255.0`。
6. 在本例子中，你應該設定 `hardware=packet`, `interrupt=10`, `ioaddr=60`。
7. 你至少要有一單獨的機器設定爲 gateway，也就是 Linux 主機:

```
name=default
host=yourlinuxhostname
hostip=192.168.1.1
gateway=1
```

8. 還要有另外一個指定領域名稱服務:

```
name=dns.domain.com ; hostip=123.123.123.123; nameserver=1
```

   注意: 用你 Linux 主機使用的 DNS 的適當資訊來取代。
9. 儲存你的 `config.tel` 檔案。
10. Telnet 到你的 Linux 主機以測試網路連接: `telnet 192.168.1.1`

3.3.6 配置執行 MacTCP 的 MacOS 機器

1. 如果你還沒爲你的乙太網路轉接器安裝適當的驅動程式，最好現在就作。
2. 打開 *MacTCP control panel*，選擇適當的網路驅動程式(Ethernet, 而非 EtherTalk) 並按下 *'More...'* 鈕。
3. 在 *'Obtain Address:'*, 按 *'Manually'*。
4. 在 *'IP Address:'* 下，從彈出選單中選擇 *class C*。忽略對話盒中的其它部份。
5. 在 *'Domain Name Server Information:'* 中填入適當資訊。
6. 在 *'Gateway Address:'* 中，填入 192.168.1.1。
7. 按下 *'OK'* 以儲存設定。在 *MacTCP control panel* 的主視窗中，在 *'IP Address:'* 盒中填入你 Mac 的 IP 位址 (192.168.1.x, 1 < x < 255)。
8. 關閉 *MacTCP control panel*。如果有的彈出視窗提醒你重新開機，那就做吧。
9. 你可以 `ping` 一下你的 Linux 主來來測試網路連線。如果你有 *MacTCP Watcher* 免費程式，按下 *'Ping'* 鈕，然後在彈出的對話盒中鍵入你的 Linux 主機的地址(192.168.1.1)。(這只不過是區域網路的連接測試，你還不能 `ping` 到外面的世界。)
10. 你可選擇性地在 System Folder 中建立一 `Hosts` 檔案以便你可以使用你區域網路裡機器的主機名稱。 這個檔案可能已經存在於你的 System Folder 裡，而且它應該會包含一些(註解掉的)範例項目而你可以根據你的需要來修改。

3.3.7 配置執行 Open Transport 的 MacOS 系統

1. 如果你還沒爲你的乙太網路轉接器安裝適當的驅動程式，最好現在就作。
2. 打開 *TCP/IP Control Panel* 然後從 *Edit* 選單中選擇 *'User Mode ...'*。確定使用者模式至少是 *'Advanced'* 然後按下 *'OK'* 鈕。
3. 從 *File* 選單中選擇 *'Configurations...'*。選擇 *'Default'* 配置並按下 *'Duplicate...'* 鈕。在 *'Duplicate Configuration'* 對話盒中鍵入 'IP Masq' (或是其它能讓你知道這是個特殊配置的字眼)，它可能會說像是 *'Deafault copy'* 什麼的。然後按下 *'OK'* 鈕，以及 *'Make Active'* 鈕。
4. 從 *'Connect via:'* 彈出式選單中選擇 *'Ethernet'*。
5. 從 *'Configure:'* 彈出式選單選擇適當的項目。如果你不知道應該選什麼，你可能應該重新選擇你的 *'Default'* 配置然後離開。我用的是 *'Manually'*。
6. 在 *'IP Address:'* 盒中輸入你的 Mac 的 IP 位址 (192.168.1.x, 1 < x < 255)。
7. 在 *'Subnet mask:'* 盒中輸入 255.255.255.0。
8. 在 *'Router address:'* 盒中輸入 192.168.1.1。
9. 在 *'Name server addr.:'* 盒中輸入你的領域名稱伺服器 IP 位址。
10. 在 *'Implicit Search Path:'* 裡的 *'Starting domain name'* 輸入你的網際網路領域名稱(例如 'microsoft.com')。

11. 接下來的步驟是選擇性的。不正確的值可能導致嚴重的錯誤行爲。如果你不確定，最好留下空白，不要勾選。如果需要的話，除去那些欄位中的任何資訊。就我目前所知沒有辦法在 TCP/IP 對話視窗中告訴系統不要使用以前選過的另一 "Hosts" 檔案。如果你知道的話，我很有興趣了解。如果你的網路需要 802.3 框架的話就勾選 *'802.3'*。

12. 按下 *'Options...'* 以確定 TCP/IP 有作用。我使用 *'Load only when needed'* 選項。如果你執行並結束 TCP/IP 應用程式許多次而未重新啓動你的機器，你將發現不選 *'Load only when needed'* 會抑制/降低你機器的記憶體管理效能。不選此項目將使 TCP/IP 協定總是被載入便於使用。如果勾選了，TCP/IP 協定會自動在需要時載入並在不需要時釋放。載入與釋放的過程可能使你機器的記憶體變的碎裂。

13. 你可以 `ping` 一下你的 Linux 主來來測試網路連線。如果你有 *MacTCP Watcher* 免費程式，按下 *'Ping'* 鈕，然後在彈出的對話盒中鍵入你的 Linux 主機的地址(192.168.1.1)。(這只不過是區域網路的連接測試，你還不能 `ping` 到外面的世界。)

14. 你可以在 System Folder 中建立一 `Hosts` 檔案以便你可以使用你區域網路裡機器的主機名稱。這個檔案可能已經或還未存在於你的 System Folder 裡。如果有的話，它應該會包含一些(註解掉的)範例項目而你可以根據你的需要來修改。如果還沒有的話，你可以從一部正在運作 MacTCP 的系統中取回，或自己建一個(它遵循 Unix 的 `/etc/hosts` 檔案格式，在 RFC 952 中描述)。一旦你建立了這個檔案，打開 *TCP/IP control panel*，按下 *'Select Hosts File...'* 鈕，然後打開 `Hosts` 檔案。

15. 關閉對話盒或從 *File* 選單中選擇 *'Close'* 或 *'Quit'* 然後按下 *'Save'* 以儲存你所做的改變。

16. 這些改變會立刻生效，但重新開機也無害。

3.3.8 配置使用 DNS 的 Novell 網路

1. 如果你還沒爲你的乙太網路轉接器安裝適當的驅動程式，最好現在就作。

2. 從 `<ftp.novell.com/pub/updates/unixconn/lwp5>` 取回 tcpip16.exe。

3. 編輯 `c:\nwclient\startnet.bat`

   : (here is a copy of mine)

```
SET NWLANGUAGE=ENGLISH
LH LSL.COM
LH KTC2000.COM
LH IPXODI.COM
LH tcpip
LH VLM.EXE
F:
```

4. 編輯 `c:\nwclient\net.cfg`

   : (將驅動程式改爲你的， i.e. NE2000)

```
Link Driver KTC2000
        Protocol IPX 0 ETHERNET_802.3
        Frame ETHERNET_802.3
        Frame Ethernet_II
```

```
        FRAME Ethernet_802.2

NetWare DOS Requester
           FIRST NETWORK DRIVE = F
           USE DEFAULTS = OFF
           VLM = CONN.VLM
           VLM = IPXNCP.VLM
           VLM = TRAN.VLM
           VLM = SECURITY.VLM
           VLM = NDS.VLM
           VLM = BIND.VLM
           VLM = NWP.VLM
           VLM = FIO.VLM
           VLM = GENERAL.VLM
           VLM = REDIR.VLM
           VLM = PRINT.VLM
           VLM = NETX.VLM

Link Support
        Buffers 8 1500
        MemPool 4096

Protocol TCPIP
        PATH SCRIPT      C:\NET\SCRIPT
        PATH PROFILE     C:\NET\PROFILE
        PATH LWP_CFG     C:\NET\HSTACC
        PATH TCP_CFG     C:\NET\TCP
        ip_address       xxx.xxx.xxx.xxx
        ip_router        xxx.xxx.xxx.xxx
```

5. 最後建立

   c:\bin\resolv.cfg

   :

   ```
   SEARCH DNS HOSTS SEQUENTIAL
   NAMESERVER 207.103.0.2
   NAMESERVER 207.103.11.9
   ```

6. 我希望這些某些使用 Novell 網路的人有幫助。還有，這對 Netware 3.1x 或 4.x 都有用。

3.3.9 配置 OS/2 Warp

1. 如果你還沒爲你的乙太網路轉接器安裝適當的驅動程式，最好現在就作。
2. 如果你還沒裝 TCP/IP 通訊協定的話現在就裝。
3. 開啓 *Programms/TCP/IP (LAN) / TCP/IP* 設定

4. 在 *'Network'* 欄位加上你的 TCP/IP 位址並設定你的 netmask (255.255.255.0)
5. 在 *'Routing'* 欄位按下 *'Add'*。將 *Type* 欄位設定為 *'default'* 並在 *'Router Address'* 欄位中鍵入你的 Linux 主機的 IP 位址 (192.168.1.1)。
6. 將 *'Hosts'* 欄位設定與你的 Linux 主機使用相同的 DNS (名稱伺服器)位址。
7. 關閉 TCP/IP 控制台。在接下來的問題中回答 yes。
8. 重新啟動你的系統
9. 你可以 ping 你的 Linux 主機以測試網路配置。在 'OS/2 命令視窗' 上鍵入 `'ping 192.168.1.1'`。如果收到 ping 封包一切就沒問題。

3.3.10 配置其它系統

相同的邏輯應可適用於配置其它的平台。查閱上述的小節的說明。 如果你有興趣寫關於其它的作業系統的配置， 請送詳細的建立指引到 *ambrose@writeme.com* 及 *dranch@trinnet.net*。

3.4 配置 IP 轉送(Forwarding)的方式

到目前為止，你應該已經安裝好核心以及其它需要的套件，也載入了你的模組。同時，其它機器的 IP 位址，閘道，以及 DNS 也該全都設定完成。

現在，唯一剩下要做的事是使用 IP 防火牆工具來轉送適當的封包給適當的機器:

> ** 這可以用許多不同的方式來達成。下列的建議與例子對我來說能用，但你可能有不同的主意，詳節部份請參考 **4.4** 節及 **ipchains(2.2.x)** / **ipfwadm(2.0.x)** 的線上手冊。 **

> ** 本節僅提供你建立 **IP** 偽裝功能所需的最少規則，一些安全的考量並未包含進去。強烈建議你花一些時間去研究適當的防火牆規則來加強安全性。 **

3.4.1 Linux 2.2.x 核心

Ipfwadm 已經無法在 **2.2.x** 版的核心中處理 **IP** 封包的偽裝規則，請改用 **ipchains**。

```
ipchains -P forward DENY
ipchains -A forward -s yyy.yyy.yyy.yyy/x -j MASQ
```

其中 x 視你的子網路而定，為下列數字之一，而 yyy.yyy.yyy.yyy 則是你的網路位址。

```
netmask         | x  | Subnet
~~~~~~~~~~~~~~~~|~~~~|~~~~~~~~~~~~~~~
255.0.0.0       | 8  | Class A
255.255.0.0     | 16 | Class B
255.255.255.0   | 24 | Class C
255.255.255.255 | 32 | Point-to-point
```

你也可以使用這種格式 yyy.yyy.yyy.yyy/xxx.xxx.xxx.xxx, 其中 xxx.xxx.xxx.xxx 指定你的子網路遮罩，如 255.255.255.0。

例如，如果我是在一個 class C 子網路上，我得輸入:

```
ipchains -P forward DENY
ipchains -A forward -s 192.168.1.0/24 -j MASQ
```

或

```
ipchains -P forward DENY
ipchains -A forward -s 192.168.1.0/255.255.255.0 -j MASQ
```

你也可以分別對每台機器設定。 例如，如果我想讓 192.168.1.2 及 192.168.1.8 能夠存取網際網路，但不允許其它機器使用的話，我得輸入:

```
ipchains -P forward DENY
ipchains -A forward -s 192.168.1.2/32 -j MASQ
ipchains -A forward -s 192.168.1.8/32 -j MASQ
```

不要把你的預設方式(policy)定爲僞裝(masquerading) — 否則可以操控他們的遞送路徑(routing) 的人將能夠直接穿過(tunnel)你的閘道，以此僞裝他們的身分!

同樣地，你可以把這些加入 /etc/rc.d/rc.local 檔案，任何一個你比較喜歡的 rc 檔案，或是在每次你需要 IP Masquerade 時手動執行之。

關於 ipchains 的詳細使用方法，請參考 *Linux IPCHAINS HOWTO* <http://metalab.unc.edu/mdw/HOWTO/IPCHAINS-HOWTO.html>

3.4.2 Linux 2.0.x 核心

```
ipfwadm -F -p deny
ipfwadm -F -a m -S yyy.yyy.yyy.yyy/x -D 0.0.0.0/0
```

或

```
ipfwadm -F -p deny
ipfwadm -F -a masquerade -S yyy.yyy.yyy.yyy/x -D 0.0.0.0/0
```

其中 x 視你的子網路而定，爲下列數字之一，而 yyy.yyy.yyy.yyy 則是你的網路位址。

```
netmask         | x  | Subnet
~~~~~~~~~~~~~~~~|~~~~|~~~~~~~~~~~~~~~
255.0.0.0       | 8  | Class A
255.255.0.0     | 16 | Class B
255.255.255.0   | 24 | Class C
255.255.255.255 | 32 | Point-to-point
```

你也可以使用這種格式 yyy.yyy.yyy.yyy/xxx.xxx.xxx.xxx, 其中 xxx.xxx.xxx.xxx 指定你的子網路遮罩，如 255.255.255.0。

例如，如果我是在一個 class C 子網路上，我得輸入:

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

因爲 bootp 請求封包沒有合法的 IP's，客户端並不知道它的位址，對於在僞裝/防火牆上執行 bootp 伺服器的人必須在 deny 之前執行下列指令:

```
ipfwadm -I -a accept -S 0/0 68 -D 0/0 67 -W bootp_clients_net_if_name -P udp
```

你也可以分別對每台機器設定。 例如，如果我想讓 192.168.1.2 及 192.168.1.8 能夠存取網際網路，但不允許其它機器使用的話，我得輸入:

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.1.2/32 -D 0.0.0.0/0
ipfwadm -F -a m -S 192.168.1.8/32 -D 0.0.0.0/0
```

常見的錯誤是像這樣的第一行指令

```
ipfwadm -F -p masquerade
```

不要把你的預設方式(policy)定爲僞裝(masquerading) — 否則可以操控他們的遞送路徑(routing) 的人將能夠直接穿過(tunnel)你的閘道，以此僞裝他們的身分!

同樣地，你可以把這些加入 `/etc/rc.d/rc.local` 檔案，任何一個你比較喜歡的 rc 檔案，或是在每次你需要 IP Masquerade 時手動執行之。

請閱讀 4.4 節有關 Ipfwadm 的詳細指引。

3.5 測試 IP Masquerade

在這些工作完成後，現在是試試看的時候了。確定你的 Linux 主機到網際網路的連線是通的。

你可以在其它機器上試著瀏覽一些'網際網路!!!' 上的網頁，看是否能見到。我建議第一次嘗試時使用 IP 位址而不要用主機名稱，因爲你的 DNS 設定有可能並不正確。

例如，你可以使用 *http://152.19.254.81/mdw/linux.html* 來存取 Linux 文件計畫網頁 http://metalab.unc.edu/mdw/linux.html

如果你看見 The Linux Documentation Project 的字樣，那麼恭喜! 它可以運作了! 接著你可以使用主機名稱試試看，然後是 telnet, ftp, RealAudio, True Speech，以及任何 IP Masquerade 支援的東西。

到目前爲止，我還不曾在上面的設定上發生過問題，而那些花下時間讓這個絕妙功能運作的人完全同意這些設定。

4 其它 IP Masquerade 的問題及軟體支援

4.1 IP Masquerade 的問題

某些協定現在無法配合 masquerading 使用，因爲它們不是假設有關埠號的一些事情，就是在位址及埠號的資料流裡編碼資料 — 後面這些協定需要在 masquerading 程式碼裡建立特定的代理程式使它們能運作。

4.2 進入系統的服務(incoming services)

Masquerading 完全不能處理外界的服務請求 (incoming services)。 只有極少方法能允許它們，但這完全與 masquerading 無關，而且實在是標準的防火牆方式。

果你並不要求高度的安全性那麼你可以簡單地重導(redirect)這些埠。 有幾種不同的方法可以做這件事 — 我使用一隻修改過的 redir 程式(我希望這隻程式很快就能從 sunsite 及其 mirrors 取得)。 如果你希望能夠對外界進入系統的服務請求有某種程度的身分驗認(authorisation) 那麼你可以在 redir 的頂層(0.7 or above) 使用 TCP wrappers 或是 Xinetd 來允許特定 IP 位址通過，或使用其它的工具。TIS 防火牆工具集是尋找工具及資訊的好地方。

更多的詳節可在 *IP Masquerade Resource* `<http://ipmasq.cjb.net/>` 找到。

將會加上一小節更多關於轉送服務的的資訊。

4.3 已支援的客户端軟體以及其它設定方面的注意事項

> ** 下面的列表將不再被維護了。可經由 **Linux IP masquerading** 運作的應用程式請參考這裡 `<http://dijon.nais.com/~nevo/masq/>` 和 ***IP Masquerade Resource*** `<http://ipmasq.cjb.net/>` 以取得進一步的細節。 **

一般說來，使用傳輸控制協定(TCP) 或是使用者定義資料協定 (UDP)的應用程式應該都能運作。 如果你有任何關於應用程式與 IP Masquerade 相容的建議，提示或問題，請拜訪由 Lee Nevo 維護的 可與 *Linux IP masquerading* 運作的應用程式 `<http://dijon.nais.com/~nevo/masq/>` 網頁。

4.3.1 可以使用的客户端軟體

一般客户端軟體

HTTP

所有有支援的平臺，瀏覽網頁

POP & SMTP

所有有支援的平臺，電子郵件軟體

Telnet

所有有支援的平臺，遠端簽入作業

FTP

所有有支援的平臺，配合 ip_masq_ftp.o 模組(不是所有站台都能配合各種客户端軟體；例如某些不能使用 ws_ftp32 觸及的站臺卻能使用 netscape 進入)

Archie

所有有支援的平臺，檔案搜尋軟體(並非所有 archie 客户端軟體都支援)

NNTP (USENET)

所有有支援的平臺，網路新聞軟體

VRML

Windows (可能所有有支援的平臺都可以)，虛擬實境瀏覽

traceroute
: 要是 UNIX 系列的平臺，某些變種可能無法運作

ping
: 所有平臺，配合 ICMP 修補檔

anything based on IRC
: 所有有支援的平臺，配合 ip_masq_irc.o 模組

Gopher client
: 所有有支援的平臺

WAIS client
: 所有有支援的平臺

多媒體客户端軟體

Real Audio Player
: Windows, 網路資料流音訊，配合載入 ip_masq_raudio 模組

True Speech Player 1.1b
: Windows, 網路資料流音訊

Internet Wave Player
: Windows, 網路資料流音訊

Worlds Chat 0.9a
: Windows, 客户－伺服端立體交談(3D chat) 程式

Alpha Worlds
: Windows, Windows, 客户－伺服端立體交談(3D chat) 程式

Powwow
: Windows, 點對點文字聲音白板通訊，如果你呼叫別人，人們可以與你交談，但是他們不能呼叫你。

CU-SeeMe
: 所有有支援的平臺，配合載入 cuseeme 模組，詳細細節請參 閱 *IP Masquerade Resource* `<http://ipmasq.cjb.net/>`

VDOLive
: Windows, 配合 vdolive 修補檔

注意: 即使不是由你呼叫別人，使用 *ipautofw* 套件某些客户端軟體像是 IPhone 以及 Powwow 可能還是可以運作(參閱 4.6 節)

其它客户端軟體

NCSA Telnet 2.3.08
DOS, 包含 telnet, ftp, ping 等等的一組套件。

PC-anywhere for windows 2.0
MS-Windows, 經由 TCP/IP 遠端遙控 PC，只有在作爲客户端而非主機端的情形下才能運作

Socket Watch
使用 ntp — 網路時間協定

Linux net-acct package
Linux, 網路帳號管理套件

4.3.2 無法使用的客户端軟體

Intel Internet Phone Beta 2
可以連上但聲音只能單向(往外)傳送

Intel Streaming Media Viewer Beta 1
無法連上伺服器

Netscape CoolTalk
無法連接對方

talk,ntalk
這將不會運作 — 需要撰寫一份核心代理程式。

WebPhone
目前無法運作(它做了不合法的位址假設)。

X
沒有測試過，但我想除非有人建立一套 X 代理程式否則它無法運作，這可能是 masquerading 程式碼之外的一個外部程式。一個讓它運作的方式是使用 ssh 作爲鏈結並且使用其內部的 X 代理功能來執行!

4.3.3 已測試過可以作爲其它機器的平臺/作業系統

- Linux
- Solaris
- Windows 95
- Windows NT (both workstation and server)
- Windows For Workgroup 3.11 (with TCP/IP package)
- Windows 3.1 (with Chameleon package)
- Novel 4.01 Server

- OS/2 (including Warp v3)
- Macintosh OS (with MacTCP or Open Transport)
- DOS (with NCSA Telnet package, DOS Trumpet works partially)
- Amiga (with AmiTCP or AS225-stack)
- VAX Stations 3520 and 3100 with UCX (TCP/IP stack for VMS)
- Alpha/AXP with Linux/Redhat
- SCO Openserver (v3.2.4.2 and 5)
- IBM RS/6000 running AIX

基本上，所有支援 TCP/IP 而且允許你指定匣道器/路由器(gateway/router)的作業系統都應該能和 IP Masquerade 一起工作。

4.4 IP 防火牆管理 (ipfwadm)

這一節提供關於 ipfwadm 更深入的使用指引。

這是一個給在固定 PPP 位址之 PPP 連線後面的防火牆/偽裝系統使用的設定。信賴(trusted) 界面為 192.168.255.1, PPP 界面已經修改過以避免犯錯 :)。我分別列出每一個進入(incoming)以及送出(outgoing)界面來抓出變更遞送路徑(stuffed routing) 以及/或是偽裝(masquerading)等等這些個 IP spoofing 技巧。同時任何沒有明確允許的東西都是禁止的!

```
#!/bin/sh
#
# /etc/rc.d/rc.firewall, 定義防火牆配置，從 rc.local 執行。
#

PATH=/sbin:/bin:/usr/sbin:/usr/bin

# 測試用，等待一段時間然後清除所有的防火牆規則。
# 如果你希望防火牆十分鐘之後自動關閉就取消下列幾行的註解。
# (sleep 600; \
# ipfwadm -I -f; \
# ipfwadm -I -p accept; \
# ipfwadm -O -f; \
# ipfwadm -O -p accept; \
# ipfwadm -F -f; \
# ipfwadm -F -p accept; \
# ) &

# 進入偽裝閘道的設定，更新以及設定拒絕的策略(policy)。事實上
# 預設的策略沒什麼關係，因為原先就希望拒絕以及記錄所有規則
ipfwadm -I -f
ipfwadm -I -p deny
```

```
# 僞裝閘道的本地(local) 界面，區域網路裡的機器，允許連往任何
# 地方
ipfwadm -I -a accept -V 192.168.255.1 -S 192.168.0.0/16 -D 0.0.0.0/0
# 僞裝閘道的遠端(remote)界面，聲稱是區域網路裡的機器，IP spoofing
# 拒絕
ipfwadm -I -a deny -V your.static.PPP.address -S 192.168.0.0/16 -D 0.0.0.0/0 -o
# 僞裝閘道的遠端界面，任何來源，允許送往固定 (permanent) PPP
# 位址
ipfwadm -I -a accept -V your.static.PPP.address -S 0.0.0.0/0 -D your.static.PPP.address/32
# 回授(loopback)界面是允許的
ipfwadm -I -a accept -V 127.0.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
# 捕捉所有規則，任何其它的進入方式都會被拒絕並記錄。可惜沒有
# 記錄用的選項但這可以代替
ipfwadm -I -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o

# 送出僞裝閘道的設定，更新以及設定拒絕的策略(policy)。事實上
# 預設的策略沒什麼關係，因爲原先就希望拒絕以及記錄所有規則
ipfwadm -O -f
ipfwadm -O -p deny
# 本地界面，允許任何來源送出至區域網路
ipfwadm -O -a accept -V 192.168.255.1 -S 0.0.0.0/0 -D 192.168.0.0/16
# 遠端界面送出至區域網路，stuffed routing，拒絕
ipfwadm -O -a deny -V your.static.PPP.address -S 0.0.0.0/0 -D 192.168.0.0/16 -o
# 區域網路的機器從遠端界面送出，stuffed masquerading，拒絕
ipfwadm -O -a deny -V your.static.PPP.address -S 192.168.0.0/16 -D 0.0.0.0/0 -o
# 區域網路的機器從遠端界面送出，stuffed masquerading，拒絕
ipfwadm -O -a deny -V your.static.PPP.address -S 0.0.0.0/0 -D 192.168.0.0/16 -o
# 任何其它遠端界面送出的東西都是允許的
ipfwadm -O -a accept -V your.static.PPP.address -S your.static.PPP.address/32 -D 0.0.0.0/0
# 回授(loopback)界面是允許的
ipfwadm -O -a accept -V 127.0.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
# 捕捉所有規則，任何其它的送出方式都會被拒絕並記錄。可惜沒有
# 記錄用的選項但這可以代替
ipfwadm -O -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o

# 僞裝閘道的轉送設定，更新以及設定拒絕的策略(policy)。事實上
# 預設的策略沒什麼關係，因爲原先就希望拒絕以及記錄所有規則
ipfwadm -F -f
ipfwadm -F -p deny
# 僞裝區域網路的機器從本地界面送出至任何地方的資料
ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/16 -D 0.0.0.0/0
# 捕捉所有規則，任何其它的轉送方式都會被拒絕並記錄。可惜沒有
# 記錄用的選項但這可以代替
ipfwadm -F -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o
```

你可以使用 -I, -O 或是 -F 來控制到某特定節點的流量。記得這些規則集是由上往下掃描的而 -a 代表 "附加(append)"到目前現有的規則集中所以任何限制必須在全域(global)規則之前出現。例如(沒測試過

) :-

用 -I 規則。可能是速度最快的但是它只能阻止區域網路裡的機器，防火牆本身仍然可以存取"禁止"的節點。當然你可能想允許這樣的組合。

```
... start of -I rules ...
# 拒絕並記錄本地界面，區域網路裡的機器通往 204.50.10.13
ipfwadm -I -a reject -V 192.168.255.1 -S 192.168.0.0/16 -D 204.50.10.13/32 -o
# 本地界面，區域網路裡的機器，允許通往任何地方
ipfwadm -I -a accept -V 192.168.255.1 -S 192.168.0.0/16 -D 0.0.0.0/0
... end of -I rules ...
```

使用 -O 規則。最慢，因為封包首先經過僞裝但這個規則阻止防火牆存取禁止的節點。

```
... start of -O rules ...
# 拒絕並記錄送出至 204.50.10.13 的資料
ipfwadm -O -a reject -V your.static.PPP.address -S your.static.PPP.address/32 -D 204.50.10.13/32 -o
# 允許任何其它遠端界面送出的東西
ipfwadm -O -a accept -V your.static.PPP.address -S your.static.PPP.address/32 -D 0.0.0.0/0
... end of -O rules ...
```

使用 -F 規則。可能比 -I 慢而這仍然只能阻止僞裝的機器(例如內部的機器)，防火牆仍然可以取得禁止的節點。

```
... start of -F rules ...
# 拒絕並記錄 PPP  界面送出從區域網路到 204.50.10.13 的資料。
ipfwadm -F -a reject -W ppp0 -S 192.168.0.0/16 -D 204.50.10.13/32 -o
# 僞裝本地界面從區域網路送出至任何地方的資料。
ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/16 -D 0.0.0.0/0
... end of -F rules ...
```

不需要有個特定的規則來允許 192.168.0.0/16 通往 204.50.11.0, 這涵蓋於全域規則中。

有一種以上的方法可以對界面設定上述規則。例如可以使用 -W eth0 來取代 -V 192.168.255.1，可以使用 -W ppp0 來取代 -V your.static.PPP.address。個人的選擇最重要。

4.5 IP 防火牆鏈 (ipchains)

這個主要是給 2.2.x 核心使用的防火牆規則管理工具(加上一個修補檔後也可在 2.0.x 上運作)。

我們會儘快更新此節，給一些使用 ipchains 的範例。

細節請參考 *Linux IP Firewalling Chains page* <http://www.rustcorp.com/linux/ipchains/> 以及 *Linux IPCHAINS HOWTO* <http://metalab.unc.edu/mdw/HOWTO/IPCHAINS-HOWTO.html>。

4.6 IP Masquerade 以及需求式撥接(Demand-Dial-Up)

1. 如果你想把網路設定成自動撥接上網際網路，那麼 *diald* demand 撥接套件將會是很棒的工具。

2. 要設定 diald, 請查看 *Setting Up Diald for Linux Page* <http://home.pacific.net.sg/harish/diald.config.html> 網頁
3. 一旦 diald 以及 IP masq 設定完成，你可以在任何客户端機器上啓動 web, telnet 或是 ftp 連線。
4. Diald 將會偵測到進入系統的請求，然後撥接到你的 ISP 並建立連線。
5. 第一次連線將會發生逾時(timeout) 的情形。如果你使用類比式的數據機那這是無可避免的。 建立數據機連結以及 PPP 連線所花費的時間將會使你的客户端軟體不耐。 如果你使用 ISDN 連線那這是可以避免的。你得做的只是結束客户端軟體現行的程序再重新啓動即可。

4.7 IPautofw 封包轉送程式

IPautofw <ftp://ftp.netis.com/pub/members/rlynch/ipautofw.tar.gz> 是一個給 Linux masquerading 使用的一般性 TCP 及 UDP 轉送程式。一般使用需要 UDP 的套件的時候，需要載入特定的 ip_masq 模組；ip_masq_raudio, ip_masq_cuseeme,... Ipautofw 以更一般化的方式運作，它將會轉送包含這些應用程式特定模組都不會轉送的任何資料流型態。如果沒有正確地管理這可能造成安全上的漏洞。

4.8 CU-SeeMe 與 Linux IP-Masquerade 的簡短說明

由 *Michael Owings* <mailto:mikey@swampgas.com> 所提供。

4.8.1 簡介

> 本節解釋讓 CU-SeeMe (包括 Cornell 及 White Pine 版本) 與 Linux 的 IP 偽裝功能一起運作的方法。

CU-SeeMe 是一個桌上視訊會議的軟體，有 Windows 及 Macintosh 兩種客户端。 一個免費的版本可從 *Cornell University* <http://cu-seeme.cornell.edu> 取得。商業的加強版可從 *White Pine Software* <http://www.wpine.com> 取得。

IP 偽裝允許一部或更多部在區域網路上的工作站"偽裝"在一部連接到網際網路的 Linux 主機之後。 區域網路內的工作站可以幾乎透通地取用網際網路，即使它沒有合法的 IP 地址。 Linux 主機重寫從內部網路到網際網路的外送封包，使它們看起來就像是從這部 Linux 機器發出的一樣。 送進來的回應封包也被改寫並轉送至內部網路的正確工作站上。 這項安排使得許多網際網路的應用程式可以透通地從內部網路的工作站上執行。 然而，對某些應用程式來說(像 CU-SeeMe), Linux 的偽裝程式需要一些小技巧的輔助才能使得封包轉送得以正確運作。 這些輔助技巧通常來自於某些特殊的核心可載入模組。 關於 IP-Masquerading 的更多資訊，請參考 *The Linux IP Masquerading Website* <http://www.indyramp.com/masq/>。

4.8.2 讓它運作

首先你需要適當地配置核心。你應該加入 IP-Masquerading 及 IP AutoForwarding 的完整支援。IP Autoforwarding 出現在 2.0.30 及其後的核心選項 – 更早的核心你需要修補檔。 參見 *Linux IP Masquerade Resource* <http://ipmasq.cjb.net> 關於 IP-Autoforwarding 的資訊指引。

接下來，你要取得最新版本的 ip_masq_cuseeme.c。 這可經由匿名 FTP 從 *ftp://ftp.swampgas.com/pub/cuseeme/ip_masq_cuseeme.c* 取得。 這個新的模組也已經加入 2.0.31

的核心之中。你可以用新的版本來取代核心中的版本。 `ip_masq_cuseeme.c` 通常位於 Linux 核心原始碼的 `net/ipv4` 目錄中。 你必須編譯並安裝此模組。

下來，你必須建立 UDP 埠號 7648-7649 的自動轉送如下:

```
ipautofw -A -r udp 7648 7649 -c udp 7648 -u
```

或

```
ipautofw -A -r udp 7648 7649 -h www.xxx.yyy.zzz
```

第一個形式允許呼叫利用埠號 7648 (主要的 cu-seeme 埠號)進出工作站。 第二個使用 ipautofw 的方式僅允許 cu-seeme 進出 www.xxx.yyy.zzz。 我較喜歡前面這種形式，因爲它更有彈性而且沒有必要指定一個特別的工作站 IP。 然而，這個方式會要求一工作站能收到呼叫前先送出呼叫。

注意這兩種方式都將客户端機器的 UDP 埠號 7648-7649 開放給外界 – 雖然這不至於有嚴重的安全危險，你還是應特別小心。

最後，以下法載入新的 `ip_masq_cuseeme` 模組:

```
modprobe ip_masq_cuseeme
```

現在你可以從你區域網路上一部被僞裝的機器上執行 CU-SeeMe 並連接到遠端的回應器(reflector)上，或另一個 CU-SeeMe 的使用者。 你應該也可以收到呼叫。注意外界的呼叫者必須使用你 Linux 匣道器的 IP, 而不是被僞裝工作站的 IP。

4.8.3 限制/警告

密碼保護的回應器 前沒有任何辦法使用。 White Pine 使用來源 IP (由客户端程式計算)在傳送前將密碼編碼保護。 因爲我們改寫了位址，回應器使用錯誤的來源 IP 來解碼，因此得到不正確的密碼。除非 White Pine 改變他們密碼編碼的策略(我已經建議過了)，或是他們願意開放他們密碼編碼的公式以便讓我能加到 `ip_masq_cuseeme`。 由於後面一個辦法的可能性很小，因此我非常鼓勵看到這份文件的人與 White Pine 聯絡並建議他們採用前一種作法。 因爲本頁的流量相當滿高的，我懷疑我們能否產生足夠的電子郵件以使此問題能進入 White Pine 的優先處理名單中。

感謝 Thomas Griwenka 提醒我這件事。

執行回應器 不能嘗試在你跑 `ip_masq_cuseeme` 及 ipautoforwarding 埠號 7648 的相同機器上執行回應器(reflector)。 這沒有用，因爲兩者都需要配置埠號 7648。 或者用另一部可直達網際網路的主機來執行，或是在執行回應器前卸下 CU-SeeMe 的客户端支援。

多 CU-SeeMe 使用者 不能一次有多位 CU-SeeMe 使用者同時在一個區域網路上面。 這是由於 CU-SeeMe 頑固地堅持使用埠號 7648, 使得在區域網路內一次僅能有一台工作站能(容易地)被重導向。

使用 -c (control port) 來呼叫 `ipautofw`, 你可以避免指定一固定的工作站位址去使用 CU-SeeMe – 第一部使用控制埠號 7648 送出任何東西的工作站會被指定爲接收來自埠號 7648-7649 的流量。 如果經過五分鐘左右這部工作站不再使用埠號 7648, 另一部工作站就可接著使用。

協助設定 CU-SeeMe 有任何評論或問題請寫信到 *mikey@swampgas.com*。 或者如果你願意的話，你可以 透過 *CU-SeeMe* 來呼叫我 `<http://www.swampgas.com/vc/vc.htm>`。

4.9 其它相關工具

我們會儘快更新此節加上更多其它僞裝相關的工具如 ipportfw 以及 masqadmin。

5 常見問題解答

如果你想到任何有用的常見問題與解答，請寫信到 *ambrose@writeme.com* 及 *dranch@trinnet.net*。 請清楚地說明問題並附上適當的解答。謝謝!

5.1 IP Masquerade 能否和動態配置的 IP 共同運作?

是的，它能和你 ISP 動態配置的 IP, 通常是由一部 DHCP 伺服器分配的位址一起運作。 只要你有一個合法的網際網路位址，它就能運作。當然，固定的 IP 一樣沒問題。

5.2 我能使用數據機，DSL, 衛星連線等來連接網際網路並使用 IP Masquerade 嗎?

當然，只要 Linux 支援該網路界面，它就能運作。

5.3 IP Masquerade 支援什麼應用程式?

要保持一份"能用的應用程式"列表是相當困難的。 不過，大部分正常的網際網路應用程式都被支援，例如瀏覽器(Netscape, MSIE 等), ftp (像 WS_FTP), Real Audio, telnet, SSH, POP3 (收信程式 - Pine, Outlook 等), SMTP(送信程式), 等等。

使用複雜協定或特殊連線方式的應用程式，例如視訊會議軟體就需要特別的輔助工具。

進一步的細節，請參考 Lee Nevo 維護的 *applications that work thru Linux IP masquerading* `<http://dijon.nais.com/~nevo/masq/>`。

5.4 我如何在 Redhat, Debian, Slackware 等使用 IP Masquerade 呢?

不論你使用何種發行套件，本文件中描述的方法都應該能適用。 某些套件可能包含了 GUI 或是特別的設定檔案能簡化配置的步驟。 我們是盡力讓這份文件寫得適用最一般化的情形。

5.5 我升級到 2.2.x 的核心，爲什麼 IP Masquerade 不能用了?

假設你的 Linux 主機已經適當地連上網際網路與你的內部網路，有幾件事你必需檢查:

- 確定你已編譯並載入必要的特性與模組。參見前面章節的說明。
- 檢查 `/usr/src/linux/Documentation/Changes` 並確定你安裝了最低需求的網路工具。
- 確定你打開了 IP 轉送機制。試著執行

```
echo "1" > /proc/sys/net/ipv4/ip_forwarding
```

- 你必須使用 *ipchains* <http://www.rustcorp.com/linux/ipchains/> 來處理 IP 偽裝及防火牆規則。
- 將整個設定配置的過程再做一次! 很多時候只是打錯字或是你忽略了一個愚蠢的錯誤。

5.6 我已經升級到 2.0.30 後的核心，爲什麼 IP Masquerade 還不能用?

假設你的 Linux 主機已經適當地連上網際網路與你的內部網路，有幾件事你必需檢查:

- 確定你已編譯並載入必要的特性與模組。參見前面章節的說明。
- 檢查 `/usr/src/linux/Documentation/Changes` 並確定你安裝了最低需求的網路工具。
- 確定你打開了 IP 轉送機制。試著執行

```
echo "1" > /proc/sys/net/ipv4/ip_forwarding
```

- 你必須使用 *ipfwadm* <http://www.xos.nl/> 來處理 IP 偽裝及防火牆規則。要在 2.0.x 使用 ipchains 的話你必須加上修補。
- 將整個設定配置的過程再做一次! 很多時候只是打錯字或是你忽略了一個愚蠢的錯誤。

5.7 我無法讓 IP Masquerade 運作! 有 Windows 平台的選擇嗎?

所以你要放棄一個可在最低硬體需求使用的免費，可靠，高效率的解決方案，而想花錢在某個需要更高硬體，低效率且不可靠的未來? (嗯，是的，我真的遇過這樣的經驗 ;-))

好吧，那是你的事。用搜尋引擎找找 MS Proxy Server, Wingate, 或看看 www.winfiles.com。 別說是我告訴你。

5.8 我檢查過我所有的設定了，仍無法讓 IP Masquerade 工作。我該怎麼辦?

- 冷靜一下。給你自己一杯咖啡並休息一下，然後試試下面的建議。
- 看一下 *IP Masquerade Mailing List Archive* <http://home.indyramp.com/lists/masq/>, 大部份你要的答案已經在等著你了。
- 把你的問題送到 IP Masquerade 通信論壇上，細節請看下一則問題。 請僅有在你無法找到答案時才這麼做。
- 在 Linux 的網路相關討論群中發問。
- 寫信到 *ambrose@writeme.com* 及 *dranch@trinnet.net*。 同時送信給我們你會有較佳的機會得到回應。David 回信的情況較佳，至於我的回應時間嘛，不予置評。
- 再次檢查你的設定 :-)

5.9 我如何加入 IP Masquerade 通信論壇?

要加入 IP Masquerade 通信論壇的話，寫信到 *masq-subscribe@indyramp.com*。

標題與信件內容將被忽略。這可使得你收到來自論壇的每一個訊息。 如果你需要的話，歡迎你使用此方式。但如果可以的話，請改用精華區(digest)。 使用精華區可減輕伺服器的負載。注意你只能使用你用來訂閱的帳號/位址發送信件。

得到更多指令的說明，寫信到 *masq-help@tori.indyramp.com*。

5.10 我想協助 IP Masquerade 的發展。我該怎麼做?

寫信到 *masq-dev-subscribe@tori.indyramp.com* (或是精華區形式，使用 *masq-dev-digest-subscribe@tori.indyramp.com*)。 加入 IP Masquerading DEVELOPERS 的通信論壇並請教那邊偉大的開發者。

別在那邊詢問與 IP Masquerade 發展無關的問題!!!!

5.11 在那裡可找到更多 IP Masquerade 的資訊?

你可在 David 及我提過的 *Linux IP Masquerade Resource* <http://ipmasq.cjb.net/> 找到更多 IP Masquerade 的資訊。參見 6.2 小節。

你也可在 Indyramp Consulting 維護的 *The Semi-Original Linux IP Masquerading Web Site* <http://www.indyramp.com/masq/> 找到更多資訊。這也是提供 IP Masquerade 通信論壇的地方。

5.12 我想翻譯本文件成其它語言，我該怎麼做?

請先確定你想翻譯的語言還沒有其它人做過。 一份可取得的翻譯文件列表可從 *Linux IP Masquerade Resource* <http://ipmasq.cjb.net/> 得到。

寫信給我到 *ambrose@writeme.com*, 我會寄最新版的 SGML 原始檔給你。

5.13 這文件有點過時了，你還有在維護嗎? 你能加進更多 ... 的資訊嗎? 有任何改善的計劃嗎?

是的，本文件有被維護。因爲兩份工作使我太忙而沒有太多時間放在這邊使我感到內疚，我很抱歉。然而，有了 David Ranch 加入本文件的維護，情況應有改善。

如果你覺得某一主題應被加到文件中，請寫信給我及 David。 最好你可以提供該資訊。如果合適的話我及 David 會把這資訊加到文件中。非常感謝你的貢獻。

我們也有一些新的點子與計劃來改進這份文件，像是包含不同網路設定的案例探討， 更多關於安全性，ipchains 的使用，ipfwadm/ipchains 規則範例，更多的常見問題與解答， 以及包含更多協定與埠號轉送的工具像是 `masqadmin` 等等。 如果你覺得你可以幫忙的話，請儘管做。謝謝。

5.14 我終於讓 IP Masquerade 能動了，好棒! 我想謝謝你們，我該怎麼做?

感謝這些發展者並感激他們所投入的時間與努力。寫信給我們並讓我們知道你有多快樂。 向其它人介紹 Linux 並幫他們解決問題。

6 其它

6.1 有用的資源

- *IP Masquerade Resource page* <http://ipmasq.cjb.net/> 應有更多建立 IP Masquerade 的資訊。
- *IP masquerade mailing list archive* <http://www.indyramp.com/masq/list/> 包含最近送到通信論壇上的信件。
- 給核心 2.2.x 與 2.0.x 的 *Linux IP Masquerade mini HOWTO* <http://ipmasq.cjb.net/ipmasq-HOWTO.html>。
- 如果你用更舊的核心，請看 *IP Masquerade HOWTO for kernel 1.2.x* <http://ipmasq.cjb.net/ipmasq-HOWTO-1.2.x.txt>
- *IP masquerade FAQ* <http://www.indyramp.com/masq/ip_masquerade.txt> 有某些一般的資訊。
- *Linux IPCHAINS HOWTO* <http://metalab.unc.edu/mdw/HOWTO/IPCHAINS-HOWTO.html> 及 *http://www.rustcorp.com/linux/ipchains/* 有許多關於 ipchains 使用的資訊，並有 ipchains 的原始碼及二進位執行檔。
- *X/OS Ipfwadm page* <http://www.xos.nl/linux/ipfwadm/> 包含 ipfwadm 套件的原始碼，執行檔，文件及其它的資訊。
- 由 Lee Nevo 所維護的網頁 可與 *Linux IP masquerading* 運作的應用程式 <http://dijon.nais.com/~nevo/masq/> 提供提示與技巧使得應用程式能與 IP Masquerade 運作。
- The *LDP Network Administrator's Guide* <http://metalab.unc.edu/mdw/LDP/nag/nag.html> 這是新手嘗試設定網路的必要資訊。
- *Trinity OS Doc* <http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS.wri>, 一份關於 Linux 網路相當容易理解的指引。
- *Linux NET-3 HOWTO* <http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html> 也有許多關於 Linux 網路方面有用的資訊。
- *Linux ISP Hookup HOWTO* <http://metalab.unc.edu/mdw/HOWTO/ISP-Hookup-HOWTO.html> 以及 *Linux PPP HOWTO* <http://metalab.unc.edu/mdw/HOWTO/PPP-HOWTO.html> 給你如何把 Linux 主機連上網際網路的相關資訊。
- *Linux Ethernet-Howto* <http://metalab.unc.edu/mdw/HOWTO/Ethernet-HOWTO.html> 有關設定執行乙太網路的區域網路很棒的資訊來源。
- 你也可能對 *Linux Firewalling and Proxy Server HOWTO* <http://metalab.unc.edu/mdw/HOWTO/Firewall-HOWTO.html> 有興趣。

- *Linux Kernel HOWTO* <http://metalab.unc.edu/mdw/HOWTO/Kernel-HOWTO.html> 將會指引你編譯核心的過程。
- 其它的 *Linux HOWTOs* <http://metalab.unc.edu/mdw/HOWTO/HOWTO-INDEX-3.html> 像是 Kernel HOWTO。
- 張貼到這個 USENET 新聞群組: *comp.os.linux.networking*

6.2 Linux IP Masquerade 資源

Linux IP Masquerade Resource <http://ipmasq.cjb.net/> 是一專門用來提供 Linux IP Masquerade 資訊的站台，也是由 David Ranch 及我來維護。這邊有關於 IP Masquerade 最新的資訊，可能也有未包含在本文件的其它資訊。

你可在下列地點找到 Linux IP Masquerade Resource:

- *http://ipmasq.cjb.net/*, 主站台，重導至 *http://www.tor.shaw.wave.ca/˜ambrose/*
- *http://ipmasq2.cjb.net/*, 第二分站, 重導至 *http://www.geocities.com/SiliconValley/Heights/2288/*

6.3 感謝

- David Ranch, dranch@trinnet.net
  協助維護本文件及 Linux IP Masquerade 資源網頁, ..., 族繁不及備載 :-)
- Michael Owings, mikey@swampgas.com
  提供關於 CU-SeeMe 的簡短說明文件。
- Gabriel Beitler, gbeitler@aciscorp.com
  on providing section 3.3.8 (setting up Novel)
- Ed Doolittle, dolittle@math.toronto.edu
  建議在 `ipfwadm` 加上 `-V` 選項以增加安全性。
- Matthew Driver, mdriver@cfmeu.asn.au
  on helping extensively on this HOWTO, and providing section 3.3.1 (setting up Windows 95)
- Ken Eves, ken@eves.com
  on the FAQ that provides invaluable information for this HOWTO
- Ed. Lott, edlott@neosoft.com
  for a long list of tested system and software
- Nigel Metheringham, Nigel.Metheringham@theplanet.net
  on contributing his version of IP Packet Filtering and IP Masquerading HOWTO, which make this HOWTO a better and technical in-depth document
  section 4.1, 4.2, and others
- Keith Owens, kaos@ocs.com.au
  on providing an excellent guide on ipfwadm section 4.2
  on correction to `ipfwadm -deny` option which avoids a security hole, and clarified the status of `ping` over IP Masquerade

- Rob Pelkey, rpelkey@abacus.bates.edu
  on providing section 3.3.6 and 3.3.7 (setting up MacTCP and Open Transport)
- Harish Pillay, h.pillay@ieee.org
  on providing section 4.5 (dial-on-demand using diald)
- Mark Purcell, purcell@rmcs.cranfield.ac.uk
  on providing section 4.6 (IPautofw)
- Ueli Rutishauser, rutish@ibm.net
  on providing section 3.3.9 (setting up OS/2 Warp)
- John B. (Brent) Williams, forerunner@mercury.net
  on providing section 3.3.7 (setting up Open Transport)
- Enrique Pessoa Xavier, enrique@labma.ufrj.br
  on the bootp setup suggestion
- developers of IP Masquerade for this great feature
  - Delian Delchev, delian@wfpa.acad.bg
  - Nigel Metheringham, Nigel.Metheringham@theplanet.net
  - Keith Owens, kaos@ocs.com.au
  - Jeanette Pauline Middelink, middelin@polyware.iaf.nl
  - David A. Ranch, trinity@value.net
  - Miquel van Smoorenburg, miquels@q.cistron.nl
  - Jos Vos, jos@xos.nl
  - Paul Russell, Paul.Russell@rustcorp.com.au
  - 還有其它我忘了提的(請讓我知道)
- 所有送回饋及建議到通信論壇上的使用者，特別是那些報告文件上的錯誤以及已支援與未支援的客户端的。
- 如果我沒有包括某些使用者送給我的資訊我感到抱歉。有如此多的建議與想法送到我這裡，但我只是沒有足夠的時間去確定或者我不小心丟了。 我正盡我最大的努力整合所有送給我的資訊到這份文件裡。感謝你的辛勞，而我也希望你能體諒我的情況。

6.4 參考資料

- Ken Eves 的 IP masquerade 常見問答集
- Indyramp Consulting 之 IP masquerade 通信論壇
- X/OS 的 Ipfwadm 網頁
- 各個關於網路的 Linux HOWTOs