Linux IP Masquerade HOWTO

David Ranch, dranch@trinnet.net; Ambrose Au, ambrose@writeme.com v1.79, 21 October 1999

이 문서는 리눅스 호스트에서 IP 마스커레이드라는 기능을 사용하는 방법을 기술하고 있다. IP 마스커레이드는 Network Address Translation(NAT)의 한 형태로, 리눅스 box에 연결된 한개의 인터넷 IP 주소를 통해서 등록된 IP 주소가 없는 내부의 컴퓨터들이 인터넷을 이용하도록 하는 기능이다.

목 차

제 1 절 소개

1.1 IP Masquerading(줄여서 IP MASQ) 에 대한 소개

(역자주: [masquerade] n, 가장 무도회, 가장, 구실 [masquerade] v, 가장 무도를 하다, 가장하다, 체하다)

이 문서는 리눅스 호스트에서 IP 마스커레이드라는 기능을 사용하는 방법을 기술하고 있다. IP 마스커레이드는 Network Address Translation(NAT)의 한 형태로, 리눅스 box에 연결된 한개의 인터넷 IP 주소를 통해서 등록된 IP 주소가 없는 내부의 컴퓨터들이 인터넷을 이용하도록 하는 기능이다. 내부의 컴퓨터들은 이더넷(Ethernet), 토큰 링(TokenRing), FDDI같은 LAN 연결이나 다이얼업 PPP(역자주: 윈도우즈의 전화접속 네트워킹), 혹은 SLIP 같은 방법을 통해서 리눅스 호스트에 연결할 수 있다. 이 문서는 이더넷(Ethernet)을 이용하는 방법을 우선적으로 다룬다.

> ***이 문서는 IBM 호환 PC에서 2.0.36이상, 2.2.9이상의 안정 커널을 사용하는 사용자들을 위해서 쓰여졌다. 1.2.x 나 1.3.x의 오래된 커널은 다루지 않고, 어떤 버젼의 커널에서는 잘못된 결과를 낼 수도 있다. IP 마스커레이드를 사용하기 이전에 새로운 안정 커널로 업그레이드하기 바란다. IP 마스커레이드를 매킨토시에서 사용하고자 한다면, Taro Fukunaga,*** *tarozax@earthlink.net* ***에게 메일을 보내서 이 HOWTO의 간략한 MkLinux용 버젼을 얻길 바란다..***

1.2 머리말, 의견 및 공로

새로운 사용자들에게는 리눅스 커널(1.2.x 이전 버젼 포함)에서 IP Masq를 설정하는 것이 매우 혼동스럽다. FAQ와 메일링 리스트가 있지만, IP Masq를 위해서 쓰여진 문서는 없었다. 그리고, 메일링 리스트에도 IP Masq를 위한 HOWTO를 요청하는 글이 있었다. 그래서, 새로운 사용자들이 출발점으로 삼을 수 있도록 이 HOWTO를 쓰기로 결정했고, 숙련된 사용자들이 추후에 더 추가할 수 있길 바란다. 이 문서에 대해 어떤 종류의 아이디어나, 수정사항들도 환영한다. 그래서 이 문서가 더 좋은 문서가 되길 바란다.

이 문서는 Ken Eves의 FAQ와 IP 마스커레이드 메일링 리스트의 수많은 메시지들을 참고하여 만들어졌다. 내가 IP Masq를 설정하는데 도움을 주고, 마침내는 이 문서를 쓰는데 영감을 준 Mr. Matthew Driver 에게 특별한 감사를 표한다. 최근에는 David Ranch가 HOWTO를 재작성했으며 ,HOWOT에 많은 section들을 추가하여 이 문서가 더욱 완벽해 지도록 했다.

수정해야 할 점이나, 정보, URL, 기타의 어떠한 의견이라도 기탄없이 `ambrose@writeme.com` 과 `dranch@trinnet.net`로 보내주기 바란다. 여러분의 참여가 이 HOWTO에 많은 도움을 줄 것이다.

이 HOWTO는 여러분이 가능한 빠른 시간안에 리눅스 IP 마스커레이드 네트웍을 작동하도록 하는데 도움을 줄 의도로 쓰여졌다. Ambrose나 David가 직업적인 저자가 아니기 때문에, 여러분은 이 문서에서 일반적이지 않거나 애초의 목적이 맞지 않는 내용을 발견 할 수 있을 것이다. 이 HOWTO에 관한 최신 정보나 기타 IP 마스커레이드에 관한 자세한 사항은 우리가 의욕적으로 관리하고 있는 web page *IP Masquerade Resource* <http://ipmasq.cjb.net/> 에서 얻을 수 있다. 여러분이 IP 마스커레이드에 대해서 기술적인 의문사항이 있다면, Amrose나 David에게 메일을 보내는 대신에 IP 마스커레이드 메일링 리스트에 참가하기 바란다. IP 마스커레이드에 관한 모든 문제점은 대부분의 유저들에게 공통된 것이고, 메일링 리스트의 누군가에게서 간단한 답을 얻을 수도 있을 것이다. 덧붙여서, Ambrose나 David로부터 답장을 받는 시간보다 메일링 리스트로부터 답을 얻는 시간이 훨씬 적게 걸릴 것이다.

이 문서의 최신 버전은 다음 사이트에서 얻을 수 있고, 그곳에서 HTML이나 postscript버전의 문서도 얻을 수 있다.

- `http://ipmasq.cjb.net/:TheIPMasqueradeResources`
- `http://ipmasq2.cjb.net/:TheIPMasqueradeResourcesMIRROR`
- `TheLinuxDocumentationProject`
- `Dranch'sLinuxpage`
- *IP Masquerade Resource Mirror Sites Listing* <http://ipmasq.cjb.net/index.html##mirror> 에서 미러 사이트를 확인할 수 있다.

1.3 판권과 포기

이 문서는 Ambrose와 David가 최선을 다한 문서로서, 옳바른 내용을 담고 있다. 그러나, 리눅스 IP 마스커레이드 기능은 사람에 의해서 개발된 것이므로, 때때로 실수나 버그등이 있을 수 있다.

이 문서에 쓰여진 정보를 사용해서 생기는, 여러분의 컴퓨터의 손상이나 어떠한 손실에 대해서도 아무도 책임을 지지 않는다.

이 문서의 정보에 의해 행해진 행위 때문에 발생된 어떤 손상도 저자는 책임지지 않는다.

No person, group, or other body is responsible for any damage on your computer(s) and any other losses by using the information on this document. i.e.

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

자, 이 상의 내용을 숙지하고... 시작해 보도록 하자..

제 2 절 배경 지식

2.1 IP 마스커레이드란 무엇인가?

IP 마스커레이드는 리눅스의 네트워킹 기능으로, 상용 방화벽(firewall)이나 네트웍 라우터(network router)에서 흔히 볼 수 있는 1 대 다(one-to-many) 방식의 NAT(Network Address Translation: 네트웍 주소

해석)와 유사하다. 예을 들어서, 어떤 리눅스 호스트가 PPP(역자주: 윈도우즈의 전화접속 네트워킹에 해당함), 이더넷(Ethernet), 기타등등의 방법으로 인터넷에 연결되어 있다면, 이 리눅스 박스에 연결된(PPP, Ethernet, 기타등등) 내부의 컴퓨터들도 IP 마스커레이드를 통해서 인터넷에 연결할 수 있다. 리눅스 IP 마스커레이드를 통하면, 내부의 컴퓨터들이 *공식적으로 할당된 IP 주소*가 없더라도 가능하다.

MASQ를 사용하면, MASQ 게이트웨이(gateway: 통로가 되는 컴퓨터)를 통해서 몇대의 컴퓨터들이 *숨어서* 인터넷을 사용할 수 있다. 즉, 인터넷에 있는 다른 컴퓨터들에게는, IP MASQ를 통해서 바깥으로 나오는 정보들은 IP MASQ Linux 서버 자체가 보내는 것처럼 보인다. 이러한 기능에 덧붙여서, IP 마스커레이드는 대단히 안전한 네트웍 환경을 제공한다. 잘 구성된 마스커레이딩 시스템과 내부 LAN의 보안을 깨는 것은, 잘 구성된 방화벽의 보안을 깨는 것 만큼이나 어렵다.

2.2 현재 상황

IP 마스커레이드는 처음 개발된지 수년이 지났고, 리눅스 커널이 2.2.x 로 들어서면서 매우 성숙해졌다. 리눅스 커널은 1.3.x 버젼부터 MASQ 기능을 자체 지원했다. 현재는 수많은 개인, 또는 상업 기관들이 훌륭하게 사용하고 있다.

웹 페이지 보기, TELNET 접속, FTP, PING, TRACEROUTE, 기타등등의 통상적인 네트웍 기능은 IP 마스커레이드를 통해서 잘 작동한다. FTP, IRC와 Real Audio와 같은 것도, 적절한 IP MASQ 모듈을 적재하면 잘 작동한다. MP3나 트루 스피치(True Speech)등의 스트리밍 오디오(streaming audio)와 같은 네트웍 관련 프로그램들도 역시 작동한다. 메일링 리스트의 어떤 동료 사용자들은 화상회의 소프트웨어에서까지 좋은 결과를 얻은 바 있다.

지원되는 전체 소프트 웨어 목록은 6.2 () section에서 확인하기 바란다.

IP 마스커레이드는 여러가지 다른 OS와 하드웨어 플랫폼을 사용하는 사용자 컴퓨터들(client machines)에게도 서버로서 잘 동작한다. MASQ 내부에서 성공적으로 동작한 시스템들은 다음과 같다 :

- Unix: Sun Solaris, *BSD, Linux, Digital UNIX, 기타등등
- Microsoft Windows 95/98, Windows NT와 Windows for Workgroups (TCP/IP 패키지가 설치된 상태)
- IBM OS/2
- MacTCP or Open Transport를 사용하는 Apple Macintosh MacOS machine들
- packet 드라이버와 NCSA Telnet 패키지를 사용하는 DOS 기반 시스템
- VAXen
- 리눅스나 NT를 사용하는 Compaq/Digital Alpha 시스템
- AmiTCP 나 AS225-stack을 사용하는 Amiga 컴퓨터까지..

리스트는 더 계속될 수 있지만 요점은 다음과 같다. TCP/IP로 통신할 수 있는 OS를 사용한다면 반드시 IP 마스커레이드와 함께 동작할 수 있어야 한다!

2.3 누가 IP 마스커레이드를 사용해서 이득을 얻는가?

- 당신이 인터넷에 연결된 리눅스 호스트를 가지고 있고,
- TCP/IP가 설치되어 있고 로컬 서브넷(local subnet)을 통해서 리눅스 호스트에 연결된 컴퓨터 몇대를 가지고 있거나,
- 당신의 리눅스 호스트가 두개 이상의 모뎀을 가지고 PPP나 SLIP서버로 동작하면 내부의 다른 컴퓨터들과 연결되어 있고,

- 그 **다른** 컴퓨터들이 공식적인 IP 주소를 할당받지 않았다면,
- 그리고 물론, ISP로부터 공식적인 IP 주소를 할당받고 리눅스를 라우터(router)로 설정하거나 외부 라우터를 구입하는등의 추가비용을 들이지 않고 그 **다른** 컴퓨터들이 인터넷을 사용하도록 하고 싶다면.

2.4 누구에게 IP 마스커레이드가 필요 없는가?

- 당신의 컴퓨터가 단독으로 설치되어 있고 인터넷에 연결되어 있거나 (그러나 단독으로 존재하더라도 방화벽을 설정하는 것은 좋은 생각일 수 있다),
- **다른** 컴퓨터들을 위해서 할당된 여러개의 IP 주소를 가지고 있다면,
- 그리고 물론, 당신이 리눅스을 사용하는 '무임 승차'라는 것을 좋아하지 않고, 오히려 같은 일을 하기 위해 비싼 대가를 지불하는 것을 더 편하게 생각한다면.

2.5 IP 마스커레이드는 어떻게 동작하는가?

>Ken Eves의 IP 마스커레이드 FAQ로부터 :

```
  가장 간단한 설치의 예는 다음 그림과 같다:

  SLIP/PPP          +------------+                        +-------------+
  ISP 제공자로      |  Linux     |          SLIP/PPP      | 다른 컴퓨터 |
 <---------- modem1|    #1      |modem2 ----------- modem3|             |
   111.222.333.444 |            |           192.168.0.100 |             |
                    +------------+                        +-------------+

   위의 그림에서, IP_MASQUERADING이 설치된 리눅스 box가 Linux #1으로 설정되어
 있고 modem1을 통한 SLIP 혹은 PPP로 인터넷에 연결되어 있다. Linux #1은
 111.222.333.444라는 IP 주소가 할당되어 있다. Linux #1은 modom2를 통해서 다른
 컴퓨터가 SLIP 혹은 PPP로 접속할 수 있도록 되어 있다.

   두번째 시스템(다른 컴퓨터: 반드시 리눅스를 사용할 필요는 없다) Linux #1으로
 SLIP 혹은 PPP 접속을 한다. 다른 컴퓨터는 공식적으로 할당된 IP 주소를 가지고
 있지 않다. 그래서 내부 주소인 192.168.0.100이라는 주소가 할당되어 있다.
 (아래 참조)

   라우팅 정보가 제대로 설정되어 있으면 IP 마스커레이드를 통해서 "다른 컴퓨터"는
 마치 인터넷에 직접 연결되어 있는 것처럼(몇가지를 제외하고) 인터넷을 사용
 할 수 있다.

Pauline Middelink에 의하면:

 "다른 컴퓨터"는 Linux #1을 게이트웨이(gateway)로 설정해야 한다는 사실을 잊지
 말아야 한다(기본 라우터(default route)인가 단지 서브넷(subnet)인가는 상관없다.)
 만약 "다른 컴퓨터"가 Linux #1을 게이트웨이로 설정하지 않는다면, Linux #1은
 proxy arp를 지원하도록 설정되어야 하는데, proxy arp에 관한 것은 이 문서의
 범주를 벗어나는 내용이다.

다음은 comp.os.linux.networking에 포스팅된 글에서 발췌한 것으로 위의 예에서의
```

이름에 맞도록 수정된 것이다:

```
 o 나는 "다른 컴퓨터"가 PPP혹은 SLIP으로 연결된 나의 Linux #1을 gateway로 인식
   하도록 하였다.
 o "다른 컴퓨터"로부터 Linux #1으로 패킷이 전달될 때, Linux #1은 그 패킷에
   새로운 발신포트번호(source port number)를 할당하고 원래의 주소는 따로
   저장해 둔다. MASQ서버는 수정된 패킷을 SLIP/PPP를 통해서 인터넷으로
   전송한다.
 o 인터넷으로부터 Linux #1으로 패킷이 되돌아올 때, Linux #1은 포트번호(port
   number)를 검사해서 "다른 컴퓨터"로부터 요청되었던 것인지 확인한다. 맞다면,
   MASQ 서버는 저장해뒀던 원래의 포트번호와 IP 주소를 인터넷으로부터 온
   패킷에 다시 할당하고 "다른 컴퓨터"로 보내준다.
 o 인터넷에서 패킷을 보낸 호스트는 이런 일이 일어나는 것을 전혀 알 수 없다.
```

IP Masquerading의 또다른 예:

아래 그림에 전형적인 예가 있다:

```
  +----------+
  |          |  Ethernet
  | A-box    |::::::
  |          |.2    : 192.168.0.x
  +----------+      :
                    :      +----------+
  +----------+      :   .1 |  Linux   |   PPP 접속
  |          |      ::::::| Masq-Gate|:::::::::::::::::::// Internet
  | B-box    |::::::      |          |  111.222.333.444
  |          |.3    :      +----------+
  +----------+      :
                    :
  +----------+      :
  |          |      :
  | C-box    |::::::
  |          |.4
  +----------+

  |                        |          |
  | <----내부 네트웍----> |          | <------외부 네트웍------>
  |                        |          |
```

이 예에서는 모두 네개의 컴퓨터가 있다. 이 경우에도 마찬가지로 오른쪽 끝에는 PPP접속을 할 수 있는 서버가 있고, 더 오른족에는 정보를 교환하고자 하는 인터넷상의 호스트들이 있다고 가정한다. 리눅스 시스템인 `Masq-Gate`가 내부 네트웍의 `A-box`, `B-box`, `C-box`를 외부의 인터넷으로 연결하도록 해주는 IP Masquerading 게이트웨이이다. 내부 네트웍은 `RFC-1918`중 한가지를 사용는데, 이 경우에는 C 클래스 네트웍인 192.168.0.0이다. 리눅스 박스가 192.168.0.1의 IP 주소를 사용하며, 다른 시스템들은 다음과 같은 주소를 갖는다:

- A-Box: 192.168.0.2
- B-Box: 192.168.0.3
- C-Box: 192.168.0.4

세 개의 컴퓨터, A-box, B-box and C-box는 TCP/IP를 사용할 수 있다면 어떤 OS를 사용하고 있더라도 상관 없다. **윈도우즈 95**, **매킨토시 MacTCP 또는 OpenTransport**나 다른 **리눅스 박스**라도 IP MASQ를 통해서 인터넷에 연결될 수 있다. 연결되는 동안, 마스커레이딩을 하는 시스템, 혹은 MASQ-gate는 내부로부터의 연결을 모두 MASQ-gate 자체에서 보내는 것처럼 전환하게 된다. MASQ는 외부로부터 신호(또는 정보)가 오면, 내부에 있는 원래의 컴퓨터로 가도록 재정렬한다. 그래서 내부 네트웍에게는 마치 인터넷에 직접 연결되어 있는 것처럼 보여지고, 마스커레이딩을 사용하고 있는지 아닌지를 구별할 수 없게 된다. 이것을 "투명한" 연결이라 한다.

NOTE: 다음 사항에 대한 자세한 것은 7 ()를 참조하기 바람:

- NAT, MASQ, proxy 서버간의 차이점.
- 패킷 방화벽이 동작하는 방법.

2.6 리눅스 2.0.x 버젼에서 IP Masqeurade를 사용하기 위한 요구사항들

**** 가장 최근의 정보는 *IP Masquerade Resource* <http://ipmasq.cjb.net/>를 참조하기 바란다. ****

- 가능한 하드웨어사양. 자세한 사항은 7.1 ()참조.
- 커널 2.0.x 소스는 http://www.kernel.org/에서 구할 수 있다.
  (레드햇 5.2와 같은 최근의 리눅스 7 ()에서는 IP 마스커레이드 모두 모듈로 지원되도록 컴파일되어 있는 커널을 제공한다. 그런 경우에는 커널을 새로 컴파일할 필요가 없다. 만약 현재 사용하는 커널을 업그레이드 하려 한다면, 관련된 다른 프로그램들도 업그레이드해야 한다.(추후에 언급됨)
- 적재가능한 커널 모듈들, 2.1.85이상 권장. http://www.pi.se/blox/modules/에서 구할 수 있다.
  (modules-1.3.57가 최저 요구사항이다)
- TCP/IP 네트웍이나 LAN 구성은 *Linux NET-3 HOWTO* <http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html>와 *Network Administrator's Guide* <http://metalab.unc.edu/mdw/LDP/nag/nag.html>에서 다루고 있다.
  TrinityOS <http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>도 확인해 보기 바란다. TrinityOS는 리눅스상에서의 네트워킹에 대한 아주 좋은 안내서이며, IP MASQ, security, DNS, DHCP, Sendmail, PPP, Diald, NFS, IPSEC기반의 VPNs, 그리고 각각의 성능에 관한 것들을 다루고 있다. 약 50개 가량의 섹션들이 있다!!
- 리눅스 호스트를 인터넷에 연결하는 것에 관한 내용은 *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>, *TrinityOS* <http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>, *Linux DHCP mini-HOWTO* <http://metalab.unc.edu/mdw/HOWTO/mini/DHCP.html>, *Linux Cable Modem mini-HOWTO* <http://metalab.unc.edu/mdw/HOWTO/mini/Cable-Modem.html>에서 확인할 수 있다.
- Ipfwadm 2.3 혹은 그 이상의 버젼은 ftp://ftp.xos.nl/pub/linux/ipfwadm/ipfwadm-2.3.tar.gz에서 구할 수 있다.
  각 프로그램의 버전 요구사항에 대한 추가정보는 *Linux IPFWADM page* <http://www.xos.nl/linux/ipfwadm/>에서 확인할 수 있다.
  - 2.0.36이상의 커널에서 IPCHAINS를 사용하려 한다면 *Willy Tarreau's IPCHAINS enabler for 2.0.36* <http://www-miaif.lip6.fr/willy/pub/linux-patches/>이나 Rusty'sIPCHAINSfor2.0.xkernels를 참고하라.

- 새로운 커널의 설정, 컴파일, 설치하는 것은 *Linux Kernel HOWTO* <http://metalab.unc.edu/mdw/HOWTO/Kernel-HOWTO.html>에서 확인할 수 있다.
- 다음에 언급된 여러가지 패치를 사용해서 IP 마스커레이드에 다른 기능들을 추가할 수도 있다:
  - TCP/IP port-forwarders 또는 re-directors: 이 툴들을 사용해서, 대개는 MASQ와 같이 동작하지 않는 프로그램들을 작동하도록 할 수 있다. 이 외에도, 외부의 인터넷 사용자들이 내부의 WWW, TELNET, SMTP, FTP(패치 필요) 등등의 서버에 연결하도록 MASQ서버를 설정할 수도 있다. 더 자세한 사항은 이 HOWTO의 6.7 () 섹션을 참조하기 바란다. 2.0.x 커널을 위한 IP Masquerading 패치 리스트:
    * Steven Clarke이 만든 IPPortForwarding(IPPORTFW) - **추천함**
    * IPAutoForward와 *a mirror* <ftp://ftp.netis.com/pub/members/rlynch/ipautofw.tar.gz> (IPAUTOFW) -
    * TCP에 관해서 *REDIR* <http://ipmasq.cjb.net/redir_0.7.orig.tar.gz> (REDIR) - 추천하지 않음
    * UDPredirector (UDPRED) - 추천하지 않음

    PORTFWed FTP:

    * 외부로부터의 FTP접속을 내부의 FTP 서버로 연결하고 싶다면 FredViles'sFTPserverpatch를 내려받아 사용하라. 이에 관한 자세한 사항은 이 HOWTO의 6.7 () 섹션에서 확인할 수 있다.

    X-Windows에서의 display 연결(forwarders):

    * *X-windows forwarding (DXCP)* <ftp://sunsite.unc.edu/pub/Linux/X11/compress/dxpc-3.7.0.tar.gz>

    MASQ를 통한 ICQ를 사용하기 위한 모듈

    * AndrewDeryabin'sICQMASQmodule

    PPTP (GRE)와 SWAN (IPSEC) VPNs의 터널링 연결(tunneling forwarders):

    * JohnHardin'sVPNMasqueradeforwarders, 혹은 오래된 패치로서 *PPTP Support* <http://ipmasq.cjb.net/ip_masq_pptp.patch.gz>.

    게임 관련 패치들:

    * Glenn Lamb의 *LooseUDP for 2.0.36+* <ftp://ftp.netcom.com/pub/mu/mumford/loose-udp-2.0.36.patch.gz> 패치. WWW 브라우저에 따라서, .gz 확장자의 화일을 자동으로 열수도 있다. 다운로드만 하기 위해서는 SHIFT키를 누른상태에서 위의 URL을 클릭하라.
      더 자세한 사항을 알려면 Dan Kegel의 *NAT Page* <http://www.alumni.caltech.edu/dank/peer-nat.html>을 확인하기 바란다. 6.3.1 () 섹션과 7 () 섹션에서 다른 정보도 확인할 수 있다.

  위의 패치들에 대한 더 많은 정보와 그외의 다른 정보들을 *IP Masquerade Resource* <http://ipmasq.cjb.net/> 에서 확인할 수 있다.

2.7 리눅스 2.2.x 버젼에서 IP Masqeurade를 사용하기 위한 요구사항들

**** 가장 최근의 정보는 *IP Masquerade Resource* <http://ipmasq.cjb.net/> 를 참고하기 바란다. ****

- 커널 2.2.x의 소스는 http://www.kernel.org/에서 구할 수 있다.
  NOTE #1: 리눅스 2.2.x 버젼중에서 2.2.11 이하의 버젼은 IPCHAINSfragmentationbug를 가지고 있

다. 이런 이유로, 강력한 IPCHAINS ruleset들을 지정하면 공격에 노출되게 된다. 커널을 업그레이드 하여 문제를 해결하기 바란다. NOTE #2: Redhat 5.2와 같은 최근의 7 ()은 2.2.x 커널을 사용할 수 없을지도 모른다. DHCP, NetUtils와 같은 툴들을 업그레이드해야 할 것이다. 더 자세한 사항은 이 HOWTO에서 언급될 것이다.

- 적재가능한 커널 모듈들, 2.1.121 이상 권장. `http://www.pi.se/blox/modules/`에서 구할 수 있다.
- TCP/IP 네트워이나 LAN 구성은 *Linux NET-3 HOWTO* `<http://metalab.unc.edu/mdw/HOWTO/NET-3-HOWTO.html>`와 *Network Administrator's Guide* `<http://metalab.unc.edu/mdw/LDP/nag/nag.html>`에서 다루고 있다.
  TrinityOS `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>`도 확인해 보기 바란다. TrinityOS는 리눅스상에서의 네트워킹에 대한 아주 좋은 안내서이며, IP MASQ, security, DNS, DHCP, Sendmail, PPP, Diald, NFS, IPSEC기반의 VPNs, 그리고 각각의 성능에 관한 것 등 약 50개 가량의 섹션들이 있다!!
- 리눅스 호스트를 인터넷에 연결하는 것에 관한 내용은 *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>`, *TrinityOS* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>`, *Linux DHCP mini-HOWTO* `<http://metalab.unc.edu/mdw/HOWTO/mini/DHCP.html>`, *Linux Cable Modem mini-HOWTO* `<http://metalab.unc.edu/mdw/HOWTO/mini/Cable-Modem.html>`에서 확인할 수 있다.
- IP Chains 1.3.9 혹은 그 이상의 버젼은 `http://www.rustcorp.com/linux/ipchains/`에서 구할 수 있다.
  각 프로그램의 버젼 요구사항에 대한 추가정보는 *Linux IP Firewalling Chains page* `<http://www.rustcorp.com/linux/ipchains/>`에서 확인할 수 있다.
- 새로운 커널을 설정, 컴파일, 설치하는 것은 *Linux Kernel HOWTO* `<http://metalab.unc.edu/mdw/HOWTO/Kernel-HOWTO.html>`에서 확인할 수 있다.
- 다음에 언급된 여러가지 패치를 사용해서 IP 마스커레이드에 다른 기능들을 추가할 수도 있다:
  - TCP/IP port-forwarders 또는 re-directors:
    - *IP PortForwarding (IPMASQADM) - 추천* `<http://juanjox.kernelnotes.org/>` 또는 예전 페이지 `mirror`.
  - ICQ MASQ module
  - `AndrewDeryabin'sICQMASQmodule`

위의 패치들에 대한 더 많은 정보와 그외의 다른 정보들을 *IP Masquerade Resource* `<http://ipmasq.cjb.net/>` 에서 확인할 수 있다.

제 3 절 IP 마스커레이드 설정

만약 당신의 네트웍에 중요한 정보가 있다면, IP 마스커레이드를 구현하기 이전에 "보안"이라는 것을 생각해 보길 바란다. 기본적으로, IP MASQ는 당신이 인터넷에 연결할 수 있도록 하는 통로이지만, 인터넷상의 누군가가 당신의 내부 네트웍으로 들어오는 통로가 될 수도 있다. 일단 IP MASQ가 동작하게 되면, IPFWADM/IPCHAINS 방화벽에 매우 강력한 정책(ruleset)을 사용할 것을 강력히 권고한다. 더 자세한 정보는 6.4 () 과 6.5 () 섹션을 참조하기 바란다.

3.1 커널에서 IP 마스커레이드를 지원하도록 컴파일 하기

만약 당신의 리눅스 배포본이 다음항목들을 지원하도록 컴파일 되어져 있고 마스커레이드에 관계된 모듈들이 컴파일되어서 제공되고 있다면 커널 컴파일을 할 필요가 없다(대부분의 배포본에 포함되어 있을 것이다.):

- IPFWADM/IPCHAINS
- IP forwarding
- IP masquerading
- IP Firewalling
- 기타 등등

당신의 배포본이 마스커레이드를 지원하는지 확실하지 않다면, 7 () 섹션이나 *IP Masquerade Resource* <http://ipmasq.cjb.net/> 에서 자세한 사항을 확인할 수 있다. 당신의 패포본이 IP 마스커레이딩을 지원하는지 알 수가 없다면, 지원하지 않는다고 생각하고 다음 단계로 넘어가라. 지원하도록 되어 있든지 아니든지 상관없이, 이 섹션에는 다른 유용한 정보들이 많이 있으므로 읽어두기를 권장한다.

3.1.1 리눅스 2.0.x 커널

필요한 소프트웨어와 패치 등은 2.5 () 섹션을 참조하기 바란다.

- 우선, 커널 소스가 필요하다.(가장 최근 버전인 2.0.36이나 그 이상 버젼)
- 만약 커널 컴파일이 처음이라도 겁먹지 말기 바란다. 실제로 해 보면, 그다지 어렵지 않고 2.5 () 섹션에 나오는 몇몇 URL에서 컴파일 방법에 대해 설명하고 있다.
- `tar xvzf linux-2.0.x.tar.gz -C /usr/src` 라고 명령하여 커널을 `/usr/src/` 에 푼다.(2.0.x는 커널 버젼) 압축을 푼 다음에, `/usr/src/linux/` 라는 디렉토리나 심볼릭 링크가 있는지 확인한다.
- 패치를 가할 것이 있으면 압축을 푼 커널 소스에 패치를 가한다. 2.0.36 이상 버젼에서는, IP 마스커레이딩을 하기 위해 특별한 패치가 필요하지는 않다. IPPORTFW, PPTP, Xwindows forwarders 와 같은 기능들은 꼭 필요하지는 않은 선택사항들이다. URL들은 2.5 () 섹션을 참조하고, 최신 정보와 그 외의 패치에 관련된 URL들은 *IP Masquerade Resources* <http://ipmasq.cjb.net/> 을 참조하길 바란다.
- 아래에 커널에 포함되어야 하는 최소한의 옵션들의 목록이 있다. 현재 설치되어 있는 네트웍 인터페이스(LAN 카드, 모뎀 등등)를 사용할 수 있도록 설정하는 것도 잊지 말아야 한다. 커널을 컴파일하는 더 자세한 방법에 대해서는 *Linux Kernel HOWTO* <http://metalab.unc.edu/mdw/HOWTO/Kernel-HOWTO.html> 와 커널 소스 디렉토리 내의 README 화일을 참조하기 바란다. 다음의 옵션들에서 ***YES인가 또는 NO인가***를 확인하기 바란다. 이 HOWTO에서 나중에 설명하는 적절한 패치를 가하지 않는다면 아래의 옵션들이 모두 보이지 않을 수도 있다:

```
* Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n/?]
  - YES: 이렇게 해야 나중에 IP 마스커레이드 기능을 선택할 수 있다.

* Enable loadable module support (CONFIG_MODULES) [Y/n/?]
  - YES: IP 마스커레이드 모듈들을 적재할 수 있도록 한다.

* Networking support (CONFIG_NET) [Y/n/?]
  - YES: 네트웍을 사용 가능하게 한다.

* Network firewalls (CONFIG_FIREWALL) [Y/n/?]
```

```
  - YES: IPFWADM 방화벽을 사용 가능하게 한다.

* TCP/IP networking (CONFIG_INET)
  - YES: TCP/IP 프로토콜을 사용 가능하게 한다.

* IP: forwarding/gatewaying (CONFIG_IP_FORWARD)
  - YES: 리눅스 네트웍 패킷 포워딩과 라우팅을 가능하게 한다.
         - IPFWADM 에 의해서 제어된다.

* IP: syn cookies (CONFIG_SYN_COOKIES) [Y/n/?]
  - YES: 기본적인 네트웍 보안을 위해서 강력히 권장한다.

* IP: firewalling (CONFIG_IP_FIREWALL) [Y/n/?]
  - YES: 방화벽 기능을 사용 가능하게 한다.

* IP: firewall packet logging (CONFIG_IP_FIREWALL_VERBOSE) [Y/n/?]
  - YES: (꼭 필요하지는 않지만 강력히 권장): 방화벽의 접근 기록을 남길 수
         있도록 한다.

* IP: masquerading (CONFIG_IP_MASQUERADE [Y/n/?]
  - YES: IP 마스커레이딩 기능을 사용하여 내부 네트웍의 특정 주소로부터의
         패킷을 주소를 변경하여 외부의 TCP/IP네트웍으로 내보내게 한다.

* IP: ipautofw masquerade support (EXPERIMENTAL) (CONFIG_IP_MASQUERADE_IPAUTOFW) [Y/n/?]
  - NO:  IPautofw 은 TCP/IP 포토를 포워딩하는 구시대적인 방법이다. 물론
         작동하기는 하지만, IPPORTFW 가 더 나은 방법이다. 그러므로 IPAUTOFW은
         추천하지 않는다.

* IP: ipportfw masq support (EXPERIMENTAL) (CONFIG_IP_MASQUERADE_IPPORTFW) [Y/n/?]
  - YES: 이 옵션을 2.0.x 커널에서 사용하기 위해서는 패치를 해야 한다.

         이 옵션을 설정하면, 인터넷에 있는 외부 컴퓨터가 마스커레이드된
         내부의 특정 컴퓨터로 직접 연결할 수 있게 된다. 이 기능은 통상적으로
         내부의 SMTP, TELNET, WWW 서버에 접근하는 데 사용된다. FTP 포트
         포워딩을 하기 위해서는 FAQ섹션에 언급되어 있는 추가적인 패치를 적용
         해야 한다. 포트 포워딩에 대한 추가적인 정보는 이 HOWTO의
         Forwards 섹션을 참조하기 바란다.

* IP: ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP) [Y/n/?]
  - YES: ICMP 패킷을 마스커레이딩할 수 있도록 한다. 꼭 필요하지 않을 수도
         있으나, ICMP 지원 없이는 많은 프로그램들이 제대로 동작하지 않을
         수 있다.

* IP: loose UDP port managing (EXPERIMENTAL) (CONFIG_IP_MASQ_LOOSE_UDP) [Y/n/?]
  - YES: 이 옵션을 2.0.x 커널에서 사용하기 위해서는 패치를 해야 한다.

         이 옵션을 통해서, 내부의 컴퓨터들에서 NAT와 같은 식으로 작동하는
         네트웍 게임들을 인터넷을 통해 즐길 수 있다. 더 자세한 사항은
         이 HOWTO의 FAQ섹션에서 확인할 수 있다.

* IP: always defragment (CONFIG_IP_ALWAYS_DEFRAG) [Y/n/?]
  - YES: 이 기능은 IP 마스커레이딩 접속을 최적화 준다. - 강력히 추천

* IP: optimize as router not host (CONFIG_IP_ROUTER) [Y/n/?]
  - YES: 이 기능은 커널의 네트웍 기능을 최적화 준다.
```

```
* IP: Drop source routed frames (CONFIG_IP_NOSR) [Y/n/?]
  - YES: 기본적인 네트웍 보안을 위해서 강력히 추천한다.

* Dummy net driver support (CONFIG_DUMMY) [M/n/y/?]
  - YES: 꼭 필요하지는 않지만, 이 옵션은 문제가 발생해서 디버깅을 할 때
         도움을 줄 것이다.

* /proc filesystem support (CONFIG_PROC_FS) [Y/n/?]
  - YES: 리눅스 네트웍 포워딩을 사용하기 위해서 필요하다.
```

NOTE: 이 옵션들은 단지 IP 마스커레딩이 동작하기 위한 요소들이다. 특정한 네트웍과 특정 하드웨어를 설정하기 위해서는 필요한 다른 옵션들을 더 선택해야 한다.

- 커널 자체를 컴파일하고 나서는, 다음과 같은 명령으로 커널의 IP 마스커레딩 관련 모듈들을 컴파일하고 설치해야 한다:

  ```
  make modules; make modules_install
  ```

- 다음에는, `/etc/rc.d/rc.local` 화일에 다음과 같이 몇줄을 추가해서 IP 마스커레이드를 사용하기 위한 스크립트를 load하도록 해야 한다. 이렇게 하면 리부팅을 할 때마다 자동적으로 IP 마스커레이딩 기능을 사용할 수 있다:

  ```
  .
  .
  .
  #rc.firewall script - Start IPMASQ and the firewall
  /etc/rc.d/rc.firewall
  .
  .
  .
  ```

3.1.2 리눅스 2.2.x 커널

필요한 소프트웨어와 패치 등은 2.6 () 섹션을 참조하기 바란다.

- 우선, 2.2.x 버전의 커널 소스가 필요하다. (최근 버전인 2.2.11이나 그 이상의 버전) NOTE #1: 리눅스 2.2.x 버전중에서 2.2.11 이하의 버전은 `IPCHAINSfragmentationbug`를 가지고 있다. 이런 이유로, 강력한 IPCHAINS ruleset들을 지정하면 공격에 노출되게 된다. 커널을 업그레이드하여 문제를 해결하기 바란다.
- 만약 커널 컴파일이 처음이라도 겁먹지 말기 바란다. 실제로 해 보면, 그다지 어렵지 않고 2.6 () 섹션에 나오는 몇몇 URL에서 컴파일 방법에 대해 설명하고 있다.
- `tar xvzf linux-2.2.x.tar.gz -C /usr/src` 라고 명령하여 커널을 `/usr/src/` 에 푼다.(2.2.x는 커널 버전) 압축을 푼 다음에, `/usr/src/linux/` 라는 디렉토리나 심볼릭 링크가 있는지 확인한다.
- 패치를 가할 것이 있으면 압축을 푼 커널 소스에 패치를 가한다. 2.2.1 이상 버젼에서는, IP 마스커레이딩을 하기 위해 특별한 패치가 필요하지는 않다. PPTP, Xwindows forwarders 와 같은 기능들은 꼭 필요하지는 않은 선택사항이다. URL들은 2.6 () 섹션을 참조하고, 최신 정보와 그외의 패치에 관련된 URL들은 *IP Masquerade Resources* `<http://ipmasq.cjb.net/>` 을 참조하길 바란다.
- 아래에 커널에 포함되어야 하는 최소한의 옵션들의 목록이 있다. 현재 설치되어 있는 네트웍 인터페이스(LAN 카드, 모뎀 등등)를 사용할 수 있도록 설정하는 것도 잊지 말아야 한다. 커널을 컴파

일하는 더 자세한 방법에 대해서는 *Linux Kernel HOWTO* <http://metalab.unc.edu/mdw/HOWTO/Kernel-HOWTO.html> 와 커널 소스 디렉토리 내의 README 화일을 참조하기 바란다. 다음의 옵션들에서 ***YES인가 또는 NO인가***를 확인하기 바란다. 이 HOWTO에서 나중에 설명하는 적절한 패치를 가하지 않는다면 아래의 옵션들이 모두 보이지 않을 수도 있다:

```
* Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n/?]
  - YES: IP 마스커레이드를 위해 꼭 필요한 것은 아니지만, 이 옵션을 선택하면
         마스커레이드 모듈을 생성하고 포트 포워딩(port forwarding)을 할 수가
         있다.

* Enable loadable module support (CONFIG_MODULES) [Y/n/?]
  - YES: IP 마스커레이드 모듈들을 적재할 수 있도록 한다.

* Networking support (CONFIG_NET) [Y/n/?]
  - YES: 네트웍을 사용 가능하게 한다.

* Packet socket (CONFIG_PACKET) [Y/m/n/?]
  - YES: 꼭 필요하지는 않지만, 이 기능은 TCPDUMP를 사용해서 IP 마스커레이딩과
         관련한 문제들을 디버깅할 수 있으므로 선택할 것을 권장한다.

* Kernel/User netlink socket (CONFIG_NETLINK) [Y/n/?]
  - YES: 꼭 필요하지는 않지만, 이 기능은 방화벽의 접근 기록을 남길 수 있도록
         한다.

* Routing messages (CONFIG_RTNETLINK) [Y/n/?]
  - NO:  이 옵션은 패킷 방화벽이 기록을 남기는 것과 아무 상관이 없다.

* Network firewalls (CONFIG_FIREWALL) [Y/n/?]
  - YES: IPCHAINS 방화벽 도구를 사용할 수 있게 한다.

* TCP/IP networking (CONFIG_INET) [Y/n/?]
  - YES: TCP/IP 프로토콜을 사용할 수 있게 한다.

* IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) [Y/n/?]
  - NO:  CONFIG_IP_ROUTE_VERBOSE 를 설정하기 위해서 필요하고 깔끔한 라우팅을
         위해서 필요하다. (ipchains/마스커레이드 와는 관계없다.)

* IP: verbose route monitoring (CONFIG_IP_ROUTE_VERBOSE) [Y/n/?]
  - YES: 이 기능은 IP 스푸핑(속임) 패킷을 제거하고 그 기록을 남기는 코드를
         사용한다면 매우 유용할 것이다.

* IP: firewalling (CONFIG_IP_FIREWALL) [Y/n/?]
  - YES: 방화벽 기능을 사용할 수 있게 한다.

* IP: firewall packet netlink device (CONFIG_IP_FIREWALL_NETLINK) [Y/n/?]
  - YES: 꼭 필요하지는 않지만, 이 기능은 방화벽의 접근 기록을 남기는 기능을
         향상시켜 줄 것이다.

* IP: always defragment (required for masquerading) (CONFIG_IP_ALWAYS_DEFRAG) [Y/n/?]
  - YES: 이 기능을 선택해야지 IP 마스커레이드와 투명한 프록시 기능을 선택할
         수 있다. 이 기능은 IP 마스커레이드 접속을 최적화 하기도 한다.

* IP: masquerading (CONFIG_IP_MASQUERADE) [Y/n/?]
  - YES: 내부 주소를 외부로 내보낼 패킷으로 변환해 주는 IP 마스커레이딩 기능을
```

```
        사용 가능하게 한다.

  * IP: ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP) [Y/n/?]
    - YES: ICMP 핑 패킷을 마스커레이드 하기 위해 사용된다. (선택하지 않더라도
           ICMP 에러 코드 자체는 마스커레이드 될 것이다.) 접속에 문제가 생겼을
           때 해결하기 위해 사용되는 중요한 기능이다.

  * IP: masquerading special modules support (CONFIG_IP_MASQUERADE_MOD) [Y/n/?]
    - YES: 꼭 필요하지는 않지만, 이 기능은 나중에 TCP/IP 포트 포워딩을 사용
           가능하게 하기 위해서 선택해야 한다. 포트 포워딩을 통해서 외부로부터
           마스커레이드되는 내부의 컴퓨터로 직접 연결할 수 있다.

  * IP: ipautofw masq support (EXPERIMENTAL) (CONFIG_IP_MASQUERADE_IPAUTOFW) [N/y/m/?]
    - NO:  IPautofw 기능은 포트 포워딩을 사용하기 위해서 사용되던 구시대적인
           방법이다. 이 기능은 프로토콜 단위의 모듈을 사용하는 것이 더 낫다.

  * IP: ipportfw masq support (EXPERIMENTAL) (CONFIG_IP_MASQUERADE_IPPORTFW) [Y/m/n/?]
    - YES: IPPORTFW를 사용가능하게 한다.

           이 옵션을 선택하면, 인터넷상의 외부의 컴퓨터들이 내부의
           마스커레이드되는 컴퓨터와 직접 통신할 수 있게 된다. 이 기능은
           통상적으로 내부의 SMTP, TELNET, WWW 서버에 접속하기 위해서 사용된다.
           FTP 포트 포워딩은 FAQ 섹션에 설명되는 추가 패치를 사용해야 한다.
           포트 포워딩에 대한 추가적인 정보는 이 HOWTO의 Forwards 섹션에서
           다루고 있다.

  * IP: ip fwmark masq-forwarding support (EXPERIMENTAL) (CONFIG_IP_MASQUERADE_MFW) [Y/m/n/?]
    - NO:  IPCHAINS로 부터 직접 IP 포워딩을 할 수 있게 한다. 현재 이 코드는
           시험용이며, 권장하는 방법은 IPMASQADM 와 IPPORTFW를 사용하는 것이다.

  * IP: optimize as router not host (CONFIG_IP_ROUTER) [Y/n/?]
    - YES: 이 기능은 커널의 네트웍 기능을 최적화 해 준다.

  * IP: GRE tunnels over IP (CONFIG_NET_IPGRE) [N/y/m/?]
    - NO:  이 기능은 꼭 필요하지는 않으며, IP 마스커레이딩을 통해서 PPTP와
           GRE 터널을 사용가능하게 한다.

  * IP: TCP syncookie support (not enabled per default) (CONFIG_SYN_COOKIES) [Y/n/?]
    - YES: 기본적인 네트웍 보안을 위해서 선택할 것을 강력히 권장한다.

  * Network device support (CONFIG_NETDEVICES) [Y/n/?]
    - YES: 리눅스의 네트웍 장치를 사용할 수 있게 한다.

  * Dummy net driver support (CONFIG_DUMMY) [M/n/y/?]
    - YES: 꼭 필요하지는 않지만, 문제가 발생했을 때 디버깅 할 때 도움이 될
           것이다.

  * /proc filesystem support (CONFIG_PROC_FS) [Y/n/?]
    - YES: 리눅스의 네트웍 포워딩 시스템을 사용하기 위해서 필요하다.
```

NOTE: 이 옵션들은 단지 IP 마스커레이딩이 동작하기 위한 요소들이다. 특정한 네트웍과 특정 하드웨어를 설정하기 위해서는 필요한 다른 옵션들을 더 선택해야 한다.

- 커널 자체를 컴파일하고 나서는, 다음과 같은 명령으로 커널의 IP 마스커레이딩 관련 모듈들을 컴파일하고 설치해야 한다:

```
make modules; make modules_install
```

- 다음에는, /etc/rc.d/rc.local 화일에 다음과 같이 몇줄을 추가해서 IP 마스커레이드를 사용하기 위한 스크립트를 load하도록 해야 한다. 이렇게 하면 리부팅을 할 때마다 자동적으로 IP 마스커레이딩 기능을 사용할 수 있다:

```
.
.
.
#rc.firewall script - Start IPMASQ and the firewall
/etc/rc.d/rc.firewall
.
.
.
```

3.2 내부 LAN에 비공식적인 내부 IP 주소를 할당하기

모든 **내부의 마스커레이드 된** 컴퓨터들에 공식적인 인터넷 주소가 할당되어져 있지 않기 때문에, 외부의 인터넷 주소와 충돌하지 않도록 그 컴퓨터들에 주소를 할당할 방법이 있어야 한다.

>IP 마스커레이드 FAQ의 원본으로부터 인용:

RFC1918 은 외부와 연결되지 않는 "개인용" 네트웍에 사용되는 IP 주소들에 관한 공식적인 문서이다. 이러한 경우에 사용되기 위해서 세 가지의 주소 영역이 있다.

```
Section 3: 개인용 주소 영역

인터넷 주소 할당 기구(The Internet Assigned Numbers Authority : IANA)는
IP 주소중에서 다음 세가지 영역을 개인용 네트웍을 위해서 예약해 두었다:

          10.0.0.0        -   10.255.255.255
          172.16.0.0      -   172.31.255.255
          192.168.0.0     -   192.168.255.255

첫번째 영역은 "24-bit 영역", 두번째는 "20-bit 영역", 세번째는 "16-bit 영역"으로
부르기로 한다. 첫번째 영역은 class A 네트웍 주소 영역이며, 두번째는 class B
네트웍 주소의 연속된 16개의 번호들이고, 세번째는 class C 네트웍 주소의 연속된
255개의 번호들이다.
```

설명을 위해서, 필자는 192.168.0.0 네트웍과 255.255.255.0의 class-C 서브넷 마스크를 사용했고, 이 HOWTO에서도 이 주소를 사용할 것이다. 그러나, 위에 있는 개인용 네트웍 주소중에서 어떤 것을 사용해도 무방하다. 단, 각각의 경우에 적절한 서브넷 마스크를 사용해야 한다.

만약 Class-C 네트웍을 사용한다면, 마스커레이딩을 사용할 컴퓨터들에 192.168.0.1, 192.168.0.2, 192.168.0.3, ..., 192.168.0.x 등과 같이 주소를 할당해야 한다.

192.168.0.1 은 보통 내부 게이트웨이 혹은 리눅스 마스커레이드 머신의 주소로서 외부로 연결되는 통로이다. 192.168.0.0과 192.168.0.255는 각각 "네트웍" 자체의 주소와 "브로드캐스트" 주소이다. (이 주소들은 예약된 주소들이다.) 이 주소들을 컴퓨터들에게 할당면, 네트웍이 제대로 동작하지 않을 것이다.

3.3 IP 포워딩 정책 설정하기

이제, 커널과 기타 필요한 패키지들이 준비되어 있어야 한다. 리눅스 마스커레이드 서버에도 모든 네트웍

IP 주소들과, 게이트웨이, DNS 주소들을 설정해야 한다. 네트웍 카드들을 설정하는 방법을 모른다면, 2.5 () 혹은 2.6 () 섹션에 언급된 HOWTO들을 참조하기 바란다.

이제 남은 것은 IP 방화벽 도구들을 설정해서 포워딩과 마스커레이딩을 하도록 하는 것이다:

> ** 설정은 여러가지 방법으로 할 수가 있지만, 필자는 다음에 예로 든 방법을 사용해서 성공했다. 하지만, 여러분은 다른 방법을 사용할 수도 있을 것이다.
>
> ** 이 섹션에서 제공하는 것은 IP 마스커레이드 기능이 작동하기 위한 최소한의 방화벽 정책이다. 일단 IP 마스커레이드가 제대로 동작하면(이 HOWTO에서 나중에 언급한다) 6.4 () 와 6.5 () 섹션에서 보안 강도가 보다 높은 정책들에 대해 알아보기 바란다. 더 자세한 사항은 IPFWADM (2.0.x) 혹은 IPCHAINS(2.2.x) man 페이지를 참조하기 바란다.

3.3.1 리눅스 2.0.x 커널

다음과 같은 "간단한" 초기 정책으로 /etc/rc.d/rc.firewall 화일을 생성한다:

```
# rc.firewall - Initial SIMPLE IP Masquerade setup for 2.0.x kernels using IPFWADM
#
# Load all required IP MASQ modules
#
#   NOTE:  Only load the IP MASQ modules you need.  All current available IP MASQ modules
#          are shown below but are commented out from loading.

# Needed to initially load modules
#
/sbin/depmod -a

# Supports the proper masquerading of FTP file transfers using the PORT method
#
/sbin/modprobe ip_masq_ftp

# Supports the masquerading of RealAudio over UDP.  Without this module,
#       RealAudio WILL function but in TCP mode.  This can cause a reduction
#       in sound quality
#
#/sbin/modprobe ip_masq_raudio

# Supports the masquerading of IRC DCC file transfers
#
#/sbin/modprobe ip_masq_irc

# Supports the masquerading of Quake and QuakeWorld by default.  This modules is
#   for for multiple users behind the Linux MASQ server.  If you are going to play
#   Quake I, II, and III, use the second example.
#
#Quake I / QuakeWorld (ports 26000 and 27000)
#/sbin/modprobe ip_masq_quake
#
#Quake I/II/III / QuakeWorld (ports 26000, 27000, 27910, 27960)
#/sbin/modprobe ip_masq_quake ports=26000,27000,27910,27960
```

```
# Supports the masquerading of the CuSeeme video conferencing software
#
#/sbin/modprobe ip_masq_cuseeme

#Supports the masquerading of the VDO-live video conferencing software
#
#/sbin/modprobe ip_masq_vdolive


#CRITICAL:  Enable IP forwarding since it is disabled by default since
#
#           Redhat Users:  you may try changing the options in /etc/sysconfig/network from:
#
#                       FORWARD_IPV4=false
#                             to
#                       FORWARD_IPV4=true
#
echo "1" > /proc/sys/net/ipv4/ip_forward

# Dynamic IP users:
#
#   If you get your Internet IP address dynamically from SLIP, PPP, or DHCP, enable this following
#       option.  This enables dynamic-ip address hacking in IP MASQ, making the life
#       with DialD, PPPd, and similar programs much easier.
#
#echo "1" > /proc/sys/net/ipv4/ip_dynaddr


# MASQ timeouts
#
#   2 hrs timeout for TCP session timeouts
#  10 sec timeout for traffic after the TCP/IP "FIN" packet is received
#  160 sec timeout for UDP traffic (Important for MASQ'ed ICQ users)
#
/sbin/ipfwadm -M -s 7200 10 160


# DHCP:  For people who receive their external IP address from either DHCP or BOOTP
#        such as ADSL or Cablemodem users, it is necessary to use the following
#        before the deny command.  The "bootp_client_net_if_name" should be replaced
#        the name of the link that the DHCP/BOOTP server will put an address on to?
#        This will be something like "eth0", "eth1", etc.
#
#        This example is currently commented out.
#
#
#/sbin/ipfwadm -I -a accept -S 0/0 67 -D 0/0 68 -W bootp_clients_net_if_name -P udp


# Enable simple IP forwarding and Masquerading
#
#  NOTE:  The following is an example for an internal LAN address in the 192.168.0.x
#         network with a 255.255.255.0 or a "24" bit subnet mask.
#
#         Please change this network number and subnet mask to match your internal LAN setup
```

```
#
/sbin/ipfwadm -F -p deny
/sbin/ipfwadm -F -a m -S 192.168.0.0/24 -D 0.0.0.0/0
```

/etc/rc.d/rc.firewall 화일을 편집해서 정책을 생성하고 나면, "`chmod 700 /etc/rc.d/rc.firewall`" 라고 명령해서 실행가능한 화일로 만든다.

위의 방법처럼 전체 TCP/IP 네트웍에 대해서가 아니라, 각각의 머신별로 IP 마스커레이딩을 설정할 수도 있다. 예를 들어서, 192.168.0.2와 192.168.0.8의 주소를 갖는 호스트는 인터넷에 접근가능하도록 하고 다른 내부의 머신들은 접근하지 못하도록 하고자 한다면, 위의 /etc/rc.d/rc.firewall 화일에서 "Enable simple IP forwarding and Masquerading" 이라고 되어 있는 부분을 바꿔주면 된다.

```
# Enable simple IP forwarding and Masquerading
#
#  NOTE:  The following is an example to only allow IP Masquerading for the 192.168.0.2
#         and 192.168.0.8 machines with a 255.255.255.0 or a "24" bit subnet mask.
#
#         Please use the following in ADDITION to the simple ruleset above for specific
#         MASQ networks.  Also change the network numbers and subnet masks to match your
#         internal LAN setup
#
/sbin/ipfwadm -F -p deny
/sbin/ipfwadm -F -a m -S 192.168.0.2/32 -D 0.0.0.0/0
/sbin/ipfwadm -F -a m -S 192.168.0.8/32 -D 0.0.0.0/0
```

IP 마스커레이딩을 처음 사용하는 사람들이 흔히 저지르는 실수는 다음과 같이 명령하는 것이다:

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

디폴트로 마스커레이딩을 하도록 해서는 **안된다**. 만약 그렇게 설정하면 라우팅 테이블을 다룰 줄 아는 어떤 누군가가 여러분의 게이트웨이를 통해서 자신의 신분을 숨기고서 어딘가로 접속할 수가 있게 된다!

위의 설정화일 내용은, `/etc/rc.d/rc.firewall` 화일이나 혹은 원하는 다른 rc 화일에 넣을 수도 있고, 아니면 IP 마스커레이드가 필요할 때마다 수동으로 명령할 수도 있다.

6.4 () 과 6.5 () 섹션에서 IPFWADM에 관한 자세한 안내와 더 강력한 IPFWADM 정책들의 예를 볼수가 있다.

3.3.2 리눅스 2.2.x 커널

2.1.x 나 2.2.x 커널에서 IP 마스커레이딩 정책들을 다루기 위한 방화벽 도구로서 **IPFWADM은 더이상 사용되지 않는다** 이 새 버전의 커널들은 이제 IPCHAINS라는 도구를 사용한다. 이렇게 된 자세한 이유는 7 () 섹션을 참조하기 바란다.

다음과 같은 "간단한" 초기 정책으로 /etc/rc.d/rc.firewall 화일을 생성한다:

```
#!/bin/sh
#
# rc.firewall - Initial SIMPLE IP Masquerade test for 2.1.x and 2.2.x kernels using IPCHAINS
#
# Load all required IP MASQ modules
#
#   NOTE:  Only load the IP MASQ modules you need.  All current IP MASQ modules
#          are shown below but are commented out from loading.
```

```
# Needed to initially load modules
#
/sbin/depmod -a

# Supports the proper masquerading of FTP file transfers using the PORT method
#
/sbin/modprobe ip_masq_ftp

# Supports the masquerading of RealAudio over UDP.  Without this module,
#       RealAudio WILL function but in TCP mode.  This can cause a reduction
#       in sound quality
#
#/sbin/modprobe ip_masq_raudio

# Supports the masquerading of IRC DCC file transfers
#
#/sbin/modprobe ip_masq_irc


# Supports the masquerading of Quake and QuakeWorld by default.  This modules is
#   for for multiple users behind the Linux MASQ server.  If you are going to play
#   Quake I, II, and III, use the second example.
#
#Quake I / QuakeWorld (ports 26000 and 27000)
#/sbin/modprobe ip_masq_quake
#
#Quake I/II/III / QuakeWorld (ports 26000, 27000, 27910, 27960)
#/sbin/modprobe ip_masq_quake ports=26000,27000,27910,27960


# Supports the masquerading of the CuSeeme video conferencing software
#
#/sbin/modprobe ip_masq_cuseeme

#Supports the masquerading of the VDO-live video conferencing software
#
#/sbin/modprobe ip_masq_vdolive


#CRITICAL:  Enable IP forwarding since it is disabled by default since
#
#           Redhat Users:  you may try changing the options in /etc/sysconfig/network from:
#
#                       FORWARD_IPV4=false
#                             to
#                       FORWARD_IPV4=true
#
echo "1" > /proc/sys/net/ipv4/ip_forward


# Dynamic IP users:
#
#   If you get your IP address dynamically from SLIP, PPP, or DHCP, enable this following
#       option.  This enables dynamic-ip address hacking in IP MASQ, making the life
```

```
#       with Diald and similar programs much easier.
#
#echo "1" > /proc/sys/net/ipv4/ip_dynaddr


# MASQ timeouts
#
#   2 hrs timeout for TCP session timeouts
#  10 sec timeout for traffic after the TCP/IP "FIN" packet is received
#  160 sec timeout for UDP traffic (Important for MASQ'ed ICQ users)
#
/sbin/ipchains -M -S 7200 10 160


# DHCP:  For people who receive their external IP address from either DHCP or BOOTP
#        such as ADSL or Cablemodem users, it is necessary to use the following
#        before the deny command.  The "bootp_client_net_if_name" should be replaced
#        the name of the link that the DHCP/BOOTP server will put an address on to?
#        This will be something like "eth0", "eth1", etc.
#
#        This example is currently commented out.
#
#
#/sbin/ipchains -A input -j ACCEPT -i bootp_clients_net_if_name -s 0/0 67 -d 0/0 68 -p udp

# Enable simple IP forwarding and Masquerading
#
#  NOTE:  The following is an example for an internal LAN address in the 192.168.0.x
#         network with a 255.255.255.0 or a "24" bit subnet mask.
#
#         Please change this network number and subnet mask to match your internal LAN setup
#
/sbin/ipchains -P forward DENY
/sbin/ipchains -A forward -s 192.168.0.0/24 -j MASQ
```

/etc/rc.d/rc.firewall 화일을 편집해서 정책을 생성하고 나면, `chmod 700 /etc/rc.d/rc.firewall`라고 명령해서 실행가능한 화일로 만든다.

위의 방법처럼 전체 TCP/IP 네트웍에 대해서가 아니라, 각각의 머신별로 IP 마스커레이딩을 설정할 수도 있다. 예를 들어서, 192.168.0.2와 192.168.0.8의 주소를 갖는 호스트는 인터넷에 접근가능하도록 하고 다른 내부의 머신들은 접근하지 못하도록 하고자 한다면, 위의 /etc/rc.d/rc.firewall 화일에서 "Enable simple IP forwarding and Masquerading" 이라고 되어 있는 부분을 바꿔주면 된다.

```
#!/bin/sh
#
# Enable simple IP forwarding and Masquerading
#
#  NOTE:  The following is an example to only allow IP Masquerading for the 192.168.0.2
#         and 192.168.0.8 machines with a 255.255.255.0 or a "24" bit subnet mask.
#
#         Please change this network number and subnet mask to match your internal LAN setup
#
/sbin/ipchains -P forward deny
```

```
/sbin/ipchains -A forward -s 192.168.0.2/32 -j MASQ
/sbin/ipchains -A forward -s 192.168.0.8/32 -j MASQ
```

IP 마스커레이딩을 처음 사용하는 사람들이 흔히 저지르는 실수는 다음과 같이 명령하는 것이다:

/sbin/ipchains -P forward masquerade

디폴트로 마스커레이딩을 하도록 해서는 **안된다**. 만약 그렇게 설정하면 라우팅 테이블을 다룰 줄 아는 어떤 누군가가 여러분의 게이트웨이를 통해서 자신의 신분을 숨기고서 어딘가로 접속할 수가 있게 된다!

위의 설정화일 내용은, `/etc/rc.d/rc.firewall` 화일이나 혹은 원하는 다른 rc 화일에 넣을 수도 있고, 아니면 IP 마스커레이드가 필요할 때마다 수동으로 명령할 수도 있다.

6.4 () 과 6.5 () 섹션에서 IPCHAINS에 관한 자세한 안내와 더 강력한 IPCHAINS 정책들의 예를 볼 수가 있다. IPCHAINS의 사용법에 관한 자세한 사항은 `LinuxIPCHAINSHOWTO`을 참조하기 바란다.

제 4 절 마스커레이딩 내부의 컴퓨터들을 설정하기

내부의 마스커레이드 되는 컴퓨터들의 IP 주소를 적절히 설정하는 것 외에, 내부의 각 컴퓨터들이 리눅스 마스커레이드 서버의 주소를 게이트웨이 주소로 설정하고 DNS 서버 주소를 적절히 설정해야 한다. 대개의 경우에 이것은 꽤 수월하다. 간단히, 게이트웨이 주소에 리눅스 호스트의 주소(일반적으로 192.168.0.1)를 입력하면 된다.

도메인 네임 서비스(DNS)의 경우에는, 사용 가능한 어떤 DNS 서버의 주소라도 추가할 수 있다. 가장 깨끗한 방법은 리눅스 서버가 사용하고 있는 DNS 서버를 입력하는 것이다. 추가로, "도메인 검색" 접미사를 추가할 수도 있다.

마스커레이드 되는 내부의 컴퓨터들을 제대로 설정하고 나면, 해당 컴퓨터의 네트웍을 재시동하든지 아니면 재부팅한다.

다음의 설정 과정에서는, 여러분이 Class C 네트웍 주소들을 사용하고, 리눅스 마스커레이드 서버의 주소가 192.168.0.1이라고 가정한다. 192.168.0.0과 192.168.0.255는 예약된 주소이니 각 컴퓨터의 주소로 사용해서는 안된다.

다음과 같은 플랫폼들이 마스커레이딩 내부에서 테스트되었다:

- Linux 1.2.x, 1.3.x, 2.0.x, 2.1.x, 2.2.x
- Solaris 2.51, 2.6, 7
- Windows 95, OSR2, 98
- Windows NT 3.51, 4.0, 2000 (웍스테이션과 서버 모두)
- Windows For Workgroup 3.11 (TCP/IP 패키지 설치)
- Windows 3.1 (Netmanage Chameleon 패키지 설치)
- TCP/IP 서비스를 설치한 Novell 4.01 서버
- OS/2 (Warp v3 포함)
- Macintosh OS (MacTCP 혹은 Open Transport 설치)
- DOS (NCSA Telnet 패키지 설치, DOS Trumpet은 부분적으로 동작)
- Amiga (AmiTCP 혹은 AS225-stack 설치)
- UCX를 설치한 VAX Stations 3520과 3100 (VMS의 경우에는 TCP/IP stack)

- Linux/Redhat을 설치한 Alpha/AXP
- SCO Openserver (v3.2.4.2와 5)
- AIX를 설치한 IBM RS/6000

4.1 Microsoft Windows 95 설정

1. 네트웍 장치 드라이버를 설치하지 않았다면 지금 설치한다. 드라이버 설치에 관한 것은 이 문서에서 다루지 않는다.
2. *'제어판'* -> *'네트웍'* 을 선택한다.
3. TCP/IP 프로토콜이 설치되지 않았다면, *추가* -> *프로토콜* -> *제조회사: Microsoft* -> *프로토콜: 'TCP/IP 프로토콜'* 을 차례로 선택해서 설치한다.
4. TCP/IP 항목을 Windows95 네트웍 카드로 연결(bound)되도록 하고 *'등록정보'*를 선책한다. *'IP 주소'* 탭을 클릭하고 IP 주소를 192.168.0.x(1 < x < 255)로 설정한다. 그리고 서브넷 마스크를 255.255.255.0으로 설정한다.
5. *"게이트웨이"* 탭을 클릭하고 *'게이트웨이'*에 192.168.0.1이라고 입력한후 "추가"를 클릭한다.
6. *'DNS 설정'* 탭을 클릭하고, 컴퓨터의 이름과 도메인 명을 입력한다. 도메인이 없다면, 여러분이 사용하는 ISP의 도메인을 입력한다. 이제, DNS 서버 주소에 리눅스 호스트가 사용하고 있는 DNS 서버(대개의 경우 `/etc/resolv.conf`화일에 저장되어 있다)를 입력한다. 이 DNS 서버들은 ISP가 운영하고 있지만, 리눅스 마스커레이드 서버에 여러분 자신의 "캐쉬"서버나 DNS 서버를 운영할 수도 있다. 원하는 도메인 검색 접미사(찾을 도메인 명)을 추가할 수도 있다.
7. 나머지 설정들은 잘 알지 못한다면 그대로 두도록 한다.
8. 모든 대화상자에서 *'확인(OK)'* 을 클릭하고 재부팅한다.
9. 네트웍 연결을 시험해 보기 위해서 리눅스 호스트로 `Ping` 을 해본다: *'시작/실행'*, `ping 192.168.0.1`라고 입력.
   (이것은 단지 내부 LAN 연결을 시험하는 것이다. 아직은 바깥 세계로 `ping` 을 할 수가 없다.) PING 한 것에 대해 응답이 없다면 네트웍 설정을 다시 확인한다.
10. C:\Windows 디렉토리에 `HOSTS` 화일을 만들면, DNS 서버가 없어도 "호스트명"으로 LAN 안에 있는 컴퓨터들에게 PING을 할 수가 있다. C:\windows 디렉토리에 `HOSTS.SAM` 라는 예제 화일이 있을 것이다.

4.2 Windows NT 설정

1. 네트웍 장치 드라이버를 설치하지 않았다면 지금 설치한다. 드라이버 설치에 관한 것은 이 문서에서 다루지 않는다.
2. *'제어판'* -> *'네트웍'* -> *프로토콜* 을 선택한다.
3. TCP/IP 서비스가 아직 설치되어 있지 않다면 *'소프트웨어 추가'* 메뉴에서 TCP/IP 프로토콜과 그외 항목들을 추가한다.
4. *'네트웍 소프트웨어와 어댑터 카드'* 부분에서, *'설치된 네트웍 소프웨어'*에 있는 *'TCP/IP 프로토콜'* 을 선택한다.
5. *'TCP/IP 설정'*에서 적절한 어댑터를 선택한다. 예를 들면 `[1]Novell NE2000 어댑터`. 그리고 IP 주소를 192.168.0.x (1 < x < 255)로 설정하고, 서브넷 마스크를 255.255.255.0, 디폴트 게이트웨이를 192.168.0.1로 설정한다.

6. Windows NT 도메인 내에 있지 않거나 각 항목이 무엇을 의미하는지 잘 모를 때에는 *'Automatic DHCP Configuration'*의 활성화를 해제하고, *'WINS Server'* 부분에 아무것도 입력하지 말고, *Enable IP Forwardings*의 활성화를 해제한다.

7. *'DNS'*를 클릭하고, 리눅스 호스트가 사용하고 있는 적절한 정보들(대개는 /etc/resolv.conf에 저장되어 있음)을 입력한다. 다 되었으면 *'확인'*을 클릭한다.

8. *'고급'*을 클릭하고, 이 옵션들이 어떤 역할을 하는 지 잘 모르면 *'DNS for Windows Name Resolution'*과 *'Enable LMHOSTS lookup'*의 활성화를 해제한다. 만약에 LMHOSTS 화일을 사용하고자 한다면, C:\winnt\system32\drivers\etc에 저장되어 있는 것을 참조한다.

9. 모든 대화상자에서 *'확인'*을 클릭하고 시스템을 재시작한다.

10. 네트웍 연결을 시험해 보기 위해서 리눅스 호스트로 `Ping` 을 해본다: *'화일/실행'*, `ping 192.168.0.1`라고 입력.
    (이것은 단지 내부 LAN 연결을 시험하는 것이다. 아직은 바깥 세계로 `ping` 을 할 수가 없다.) PING 한 것에 대해 응답이 없다면 네트웍 설정을 다시 확인한다.

4.3 Windows에서 Workgroup 3.11 설정

1. 네트웍 장치 드라이버를 설치하지 않았다면 지금 설치한다. 드라이버 설치에 관한 것은 이 문서에서 다루지 않는다.

2. TCP/IP 32b 패키지가 아직 설치되어 있지 않다면 설치한다.

3. *'Main'/'Windows Setup'/'Network Setup'*에서, *'Drivers'*를 클릭한다.

4. *'Network Drivers'* 부분에서 *'Microsoft TCP/IP-32 3.11b'*를 선택한다. *'Setup'*을 클릭한다.

5. IP 주소를 192.168.0.x (1 < x < 255)로 설정한다. 그리고 서브넷 마스크를 255.255.255.0으로 디폴트 게이트웨이를 192.168.0.1로 설정한다.

6. Windows NT 도메인 내에 있지 않거나 각 항목이 무엇을 의미하는지 잘 모를 때에는 *'Automatic DHCP Configuration'*의 활성화를 해제하고, *'WINS Server'* 부분에 아무것도 입력하지 말도록 한다.

7. *'DNS'*를 클릭하고, 리눅스 호스트가 사용하고 있는 적절한 정보들(대개는 /etc/resolv.conf에 저장되어 있음)을 입력한다. 다 되었으면 *'확인'*을 클릭한다.

8. *'고급'*을 클릭하고, *'Enable DNS for Windows Name Resolution'*과 *'Enable LMHOSTS lookup'*를 선택한다. 만약에 LMHOSTS 화일을 사용하고자 한다면, C:\winnt\system32\drivers\etc에 저장되어 있는 것을 참조한다.

9. 모든 대화상자에서 *'확인'*을 클릭하고 시스템을 재시작한다.

10. 네트웍 연결을 시험해 보기 위해서 리눅스 호스트로 `Ping` 을 해본다: *'화일/실행'*, `ping 192.168.0.1`라고 입력.
    (이것은 단지 내부 LAN 연결을 시험하는 것이다. 아직은 바깥 세계로 `ping` 을 할 수가 없다.) PING 한 것에 대해 응답이 없다면 네트웍 설정을 다시 확인한다.

4.4 UNIX 기반 시스템의 설정

1. 아직 네트웍 카드를 설치하지 않았거나 해당 드라이버를 지원하도록 커널을 다시 컴파일 하지 않았다면 지금 한다. 이 문서에서 이 내용은 다루지 않는다.

2. TCP/IP 네트웍이 아직 설치되어 있지 않다면, net-tools 패키지와 같은 TCP/IP 네트웍 툴을 설치한다.

3. *IPADDR*를 192.168.0.x (1 < x < 255)로 설정한다. *NETMASK*를 255.255.255.0, *GATEWAY*를 192.168.0.1, 그리고 *BROADCAST*를 192.168.0.255로 설정한다. 예를 들어서 래드햇 리눅스 시스템이라면, `/etc/sysconfig/network-scripts/ifcfg-eth0`화일을 편집하거나, 간단하게 Control Panel에서 해결할 수 있다. SunOS, BSDi, Slackware Linux, Solaris, SuSe, Debian 등등.. 다른 UNIX에서는 방법이 다를 수도 있다. 정보를 더 얻고자 한다면 여러분의 해당 UNIX 문서를 참조하기 바란다.

4. `/etc/resolv.conf`화일에 도메인 네임 서비스(DNS)를 추가하고 도메인 검색 접미사를 추가한다. UNIX 버젼과 종류에 따라서는, /etc/nsswitch.conf 화일을 편집해서 DNS 서비스를 사용가능하게 한다.

5. 설정에 따라서는 `/etc/networks` 화일을 편집해서 바꾸어줘야 할 수도 있다.

6. 적절한 서비스들을 재시동하거나, 혹은 간단하게 아예 시스템 자체를 재시작한다.

7. 게이트웨이가 되는 컴퓨터로의 연결을 시험하기 위해서 다음과 같이 ping 명령을 내린다: `ping 192.168.0.1`.
   (이것은 단지 내부 LAN 연결을 시험하는 것이다. 아직은 바깥 세계로 `ping` 을 할 수가 없다.) PING 한 것에 대해 응답이 없다면 네트웍 설정을 다시 확인한다.

4.5 NCSA 텔넷 패키지를 사용하는 DOS의 설정

1. 아직 네트웍 카드를 설치하지 않았다면 지금 설치한다. 네트웍 카드 설치에 관한 것은 이 문서에서 다루지 않는다.

2. 적절한 패킷 드라이버를 로드한다. 예를 들어서: NE2000 이더넷 카드를 I/O 포트 300, IRQ 10으로 사용한다면, `nwpd 0x60 10 0x300`라고 명령한다.

3. 새로운 디렉토리를 만들고, 그 디렉토리에 NCSA 텔넷 패키지를 풀어 놓는다: `pkunzip tel2308b.zip`

4. 텍스트 에디터로 `config.tel` 화일을 연다.

5. `myip=192.168.0.x` (1 < x < 255)로, netmask=255.255.255.0로 설정한다.

6. 이 예에서는, `hardware=packet, interrupt=10, ioaddr=60`라고 설정해야 한다.

7. 게이트 웨이로서 적어도 한개의 컴퓨터에 대한 설정이 있어야 한다(예를 들면 이 경우에는 리눅스 호스트):

```
name=default
host=리눅스호스트이름
hostip=192.168.0.1
gateway=1
```

8. 도메인 네임 서비스를 위해서 또 하나의 설정을 해줘야 한다:

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

   Note: 리눅스 호스트가 사용하고 있는 정보대로 위의 내용을 수정해 준다.

9. `config.tel` 화일을 저장한다.

10. 네트웍 연결을 시험하기 위해서 리눅스 호스트로 텔넷 연결을 해 본다: `telnet 192.168.0.1` 만약 로긴 프롬프트가 나오지 않는다면, 네트웍 설정을 다시 확인한다.

4.6 MacTCP를 사용하는 MacOS 기반 시스템의 설정

1. 이더넷 어댑터를 위한 소프트웨어를 아직 설치하지 않았다면 지금 설치한다. 어댑터 설치에 관한 것은 이 문서에서 다루지 않는다.

2. *MacTCP control panel*을 연다. 적절한 네트웍 드라이버를 선택한다(EtherTalk이 아니고 Ethernet을 선택). 그리고 *'More...'* 버튼을 클릭한다.

3. *'Obtain Address:'* 부분에서, *'Manually'*를 클릭한다.

4. *'IP Address:'* 부분에서, 팝업 메뉴에서 *class C*를 선택한다. 이 대화상자의 나머지 부분은 무시한다.

5. *'Domain Name Server Information:'*에서 DNS 정보를 입력한다.

6. *'Gateway Address:'*에서, 192.168.0.1를 입력한다.

7. *'OK'*를 클릭해서 설정을 저장한다. *MacTCP control panel*의 메인 윈도우에서, *'IP Address:'*부분에 Mac 컴퓨터의 IP 주소 (192.168.0.x, $1 < x < 255$)를 입력한다.

8. *MacTCP control panel*을 닫는다. 재시작을 묻는 대화상자가 나타나면 시스템을 재시작한다.

9. 네트웍 연결을 시험하기 위해서 리눅스 호스트에 ping을 보낼 수 있다. *MacTCP Watcher*라는 프리웨어 소프트웨어가 설치되어 있다면, *'Ping'* 버튼을 누르고, 나타나는 대화상자에서 리눅스 호스트의 주소(192.168.0.1)를 입력한다. (이것은 단지 내부 LAN 연결을 시험하는 것이다. 아직은 바깥 세계로 `ping` 을 할 수가 없다.) PING 한 것에 대해 응답이 없다면 네트웍 설정을 다시 확인한다.

10. 내부 LAN에서 IP 주소 대신 호스트 이름을 사용하기 위해서, 시스템 폴더에 `Hosts` 화일을 만들 수도 있다. 시스템 폴더에는 이 화일이 이미 존재하고 있을 것이다. 이 화일에는 여러분의 경우에 맞춰서 수정해서 사용할 수있는 샘플이 들어 있을 것이다.

4.7 Open Transport를 사용하는 MacOS 기반 시스템의 설정

1. 이더넷 어댑터를 위한 적절한 드라이버가 아직 설치되어 있지 않다면, 지금 설치한다. 드라이버 설치에 관한 것은 이 문서에서 다루지 않는다.

2. *TCP/IP Control Panel*을 열어서 *Edit* 메뉴에서 *'User Mode ...'*를 선택한다. user mode 가 최소한 *'Advanced'* 이상에 설정되어 있는지 확인하고 *'OK'* 버튼을 누른다.

3. *File* 메뉴에서 *'Configurations...'*를 선택한다. *'Default'* 로 되어 있는 설정을 선택하고 *'Duplicate...'* 버튼을 클릭한다. *'Duplicate Configuration'* 대화상자에서, 'IP Masq' (혹은 보통의 경우에서의 설정이 아니라는 것을 의미하는 것)을 입력한다. 이렇게 하면 *'Default copy'*와 같은 것이 나타날 것이다. 그러면 *'OK'* 버튼을 클릭하고, *'Make Active'* 버튼을 클릭한다.

4. *'Connect via:'* 대화상자에서 *'Ethernet'*을 선택한다.

5. *'Configure:'* 대화상자에서 적절한 항목을 선택한다. 어떤 것을 선택해야 하는지 잘 모르겠다면, 아마도 *'Default'* 설정을 다시 선택하고 나가야 할 것이다. 필자의 경우에는 *'Manually'* 를 선택해서 설정했다.

6. *'IP Address:'* 대화상자에서 Mac 컴퓨터의 IP 주소(192.168.0.x, $1 < x < 255$)를 입력한다.

7. *'Subnet mask:'* 대화상자에서 255.255.255.0를 입력한다.

8. *'Router address:'* 대화상자에서 192.168.0.1을 입력한다.

9. *'Name server addr.:'* 대화상자에서 DNS 서버의 IP 주소를 입력한다.

10. *'Implicit Search Path:'* 부분의 *'Starting domain name'* 대화상자에서 인터넷 도메인을 입력한다.

11. 다음 과정은 선택적이다. 잘못 설정하면 정상 동작하지 않을 수도 있다. 잘 모르겠다면, 아무것도 입력하지 말고 비워두거나, 아무것도 선택되지 않은 채로 두는 것이 좋을 것이다. 필요하다면, 입력되어 있는 정보를 모두 없애도록 한다. 필자가 아는 한, TCP/IP 대화상자를 통해서 시스템이 이전에 선택되어 있는 다른 "Hosts" 화일을 사용하지 않도록 하는 방법은 없다. 만약 여러분이 그 방법을 알고 있다면, 필자에게 알려주길 바란다. 여러분의 네트웍이 802.3 타입의 프레임을 필요로 한다면 *'802.3'*을 체크한다.

12. *'Options...'* 버튼을 클릭해서 TCP/IP 가 활성화 되도록 한다. 필자의 경우에는 *'Load only when needed'* 옵션을 사용했다. 여러분이 컴퓨터를 재부팅 하지 않는채로 TCP/IP 응용프로그램의 실행과 종료를 반복을 여러번 반복하는 경우에는, *'Load only when needed'* 옵션을 체크하지 않는 것이 여러분의 컴퓨터의 메모리 관리에 도움을 줄 것이다. 그 항목을 체크하지 않은 채로 두면, TCP/IP 프로토콜 스택은 항상 로드되어서 사용 가능한 상태가 된다. 만약 체크해 둔다면, TCP/IP 스택은 필요할 때 자동적으로 로드되고 필요없어지면 자동적으로 제거된다. 이렇게 되면, 계속적인 loading과 unloading을 함으로써 메모리가 파편으로 조작나게 된다. (역자주: 윈도우즈의 디스크 조각 모임이 필요한 이유와 비슷하다.)

13. 네트웍 연결을 시험하기 위해서 리눅스 호스트에 ping을 보낼 수 있다. *MacTCP Watcher*라는 프리웨어 소프트웨어가 설치되어 있다면, *'Ping'* 버튼을 누르고, 나타나는 대화상자에서 리눅스 호스트의 주소(192.168.0.1)를 입력한다. (이것은 단지 내부 LAN 연결을 시험하는 것이다. 아직은 바깥 세계로 `ping` 을 할 수가 없다.) PING 한 것에 대해 응답이 없다면 네트웍 설정을 다시 확인한다.

14. 내부 LAN에서 IP 주소 대신 호스트 이름을 사용하기 위해서, 시스템 폴더에 `Hosts` 화일을 만들 수도 있다. 이 화일은 시스템 폴더에 있을 수도 있고 없을 수도 있다. 이 화일이 존재한다면, 여러분의 경우에 맞춰서 수정해서 사용할 수있는 샘플이 들어 있을 것이다. 존재하지 않는다면, MacTCP를 사용하고 있는 시스템으로부터 복사해 와서 수정해 쓰거나, 그냥 여러분 자신이 만들어서 사용해도 된다(이 화일의 형식은 UNIX의 `/etc/hosts` 화일 형식의 일부를 사용하며, RFC952에 설명되어 있다). 일단 화일을 만들고 나면, *TCP/IP control panel*을 열어서, *'Select Hosts File...'* 버튼을 누르고 `Hosts` 화일을 연다.

15. 닫기 상자를 클릭하거나 *File* 메뉴에서 *'Close'* 혹은 *'Quit'* 을 선택한다. 그리고 *'Save'* 버튼을 클릭해서 변경사항을 저장한다.

16. 변경 사항은 즉시 반영되지만, 시스템을 재부팅하는 것도 좋다.

4.8 DNS를 사용하는 Novell 네트웍의 설정

1. 이더넷 어댑터을 위한 드라이버 소프트웨어가 아직 설치되어 있지 않다면, 지금 설치한다. 드라이버 설치에 관한 것은 이 문서에서 다루지 않는다.

2. *The Novell LanWorkPlace page* `<ftp.novell.com/pub/updates/unixconn/lwp5>`에서 tcpip16.exe를 다운로드한다.

3. `c:\nwclient\startnet.bat`를 편집한다

   : (필자가 쓰는 화일 내용이다)

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

4. c:\nwclient\net.cfg를 편집한다
   : (link driver는 여러분에 맞게 수정한다. 예를 들면 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      192.168.0.xxx
        ip_router       192.168.0.1

위의 "ip_address" 부분은 여러분의 IP 주소로 한다 (192.168.0.x, 1 < x < 255)
그리고 마지막으로 c:\bin\resolv.cfg를 편집한다:

SEARCH DNS HOSTS SEQUENTIAL
NAMESERVER xxx.xxx.xxx.xxx
NAMESERVER yyy.yyy.yyy.yyy
```

5. 위의 "NAMESERVER" 부분은 여러분이 사용하는 DNS 서버로 대체한다.

6. 게이트웨이 컴퓨터로의 네트웍 연결을 시험하기 위해서 ping 명령을 한다: `ping 192.168.0.1` (이것은 단지 내부 LAN 연결을 시험하는 것이다. 아직은 바깥 세계로 ping 을 할 수가 없다.) PING 한 것에 대해 응답이 없다면 네트웍 설정을 다시 확인한다.

4.9 OS/2 Warp의 설정

1. 이더넷 어댑터를 위한 적절한 드라이버 소프트웨어가 아직 설치되어 있지 않다면 지금 설치한다. 드라이버 설치에 관한 것은 이 문서에서 다루지 않는다.
2. TCP/IP 프로토콜이 아직 설치되어 있지 않다면 지금 설치한다.
3. *Programs/TCP/IP (LAN) / TCP/IP* 설정으로 간다.
4. *'Network'*에서 TCP/IP 주소(192.168.0.x)를 입력하고 네트웍 마스크를 설정한다(255.255.255.0).
5. *'Routing'*에서 *'Add'*를 누른다. *Type*은 *'default'*로 하고 *'Router Address'* 부분에서 리눅스 호스트의 IP 주소를 입력한다(192.168.0.1).
6. *'Hosts'*에 리눅스 호스트가 사용하고 있는 DNS (Nameserver) 주소와 같이 설정해 준다.
7. TCP/IP control panel을 닫고 뒤이어 나오는 질문들에 yes를 답한다.
8. 시스템을 재부팅한다.
9. 네트웍 설정을 시험하기 위해서 리눅스 호스트를 ping 할 수도 있다. 'OS/2 Command prompt Window'에서 `'ping 192.168.0.1'`라고 입력한다. ping 패킷이 돌아오면 모든게 제대로 설정된 것이다.

4.10 그외 다른 시스템의 설정

그외 다른 시스템을 설정할 때에도 지금까지와 같은 방식을 사용한다. 위의 섹션들을 참조하기 바란다. 위에서 다루지 있지 않는 시스템에서의 설정에 대해서 글을 써 주실 분은, 그 자세한 설정과정을 `ambrose@writeme.com`과 `dranch@trinnet.net`으로 보내주기 바란다.

제 5 절 IP 마스커레이드의 시험

이제 마지막으로, IP 마스커레이딩을 시험할 때다. 리눅스 호스트를 아직 재부팅해보지 않았다면, 지금 재부팅하고 부팅이 성공하는지 확인하고, /etc/rc.d/rc.firewall 정책을 실행한다. 다음으로, 내부 LAN과의 연결과 외부 인터넷으로의 연결이 제대로 되어 있는지 확인한다.

이제 다음과 같이 한다:

- 첫번째: 마스커레이딩 내부의 컴퓨터에서, 내부의 다른 컴퓨터로 ping을 해본다(예를 들면 *ping 192.168.0.10* 과 같이). 이것으로 내부 LAN의 컴퓨터에서 TCP/IP가 제대로 동작하는지 확인할 수 있다. 만약 제대로 되지 않는다면, 내부 컴퓨터들에서 TCP/IP 설정을 이 HOWTO에서 설명한 대로 제대로 했는지 다시 확인한다.
- 두번째: 마스커레이드 서버 자체에서, 마스커레이드 내트웍 내부의 IP 주소로 ping을 해본다(예를 들면 *ping 192.168.0.1*과 같이). 이제 인터넷 상의 외부 IP 주소로 ping을 해본다. 이 외부의 주소는 ISP에 연결된 자기 자신의 PPP, 이더넷 등의 주소여도 된다. 이 IP 주소를 모른다면, *"/sbin/ifconfig"*라고 명령해서 인터넷 주소를 알아낸다. 이것으로 마스커레이드 서버에 네트웍이 온전히 연결되어 있는지 알 수가 있다.
- 세번째: 다시 마스커레이딩 되는 내부의 컴퓨터에서, 마스커레이딩 리눅스 호스트의 내부 이더넷 카드에 연결된 IP 주소로 ping을 해본다(예를 들면 *ping 192.168.0.1*과 같이). 이것으로 내부 네트웍과 라우팅이 제대로 되고 있는지 확인할 수 있다. 만약 이것이 실패한다면, 마스커레이드 서버와 내부 컴퓨터의 이더넷 카드가 "물리적으로" 연결되어 있는지 확인한다. 이는 이더넷 카드 뒷면이나 이더넷 허브/스위치(만약 있다면)의 LED가 점등하는지 확인하면 된다.

- 네번째: 내부의 컴퓨터에서, 마스커레이드 서버의 외부로 연결된 TCP/IP 주소로 ping을 해본다. 이 주소는 ISP에 연결된 여러분의 PPP, 이더넷 등의 주소일 것이다. 이 ping 테스트로, 마스커레이딩(특히 ICMP 마스커레이딩)이 제대로 작동하고 있는지 확인할 수 있다. 만약 이것이 제대로 동작하지 않는다면, 커널이 "ICMP Masquerading"을 지원하도록 되어 있는지와 /etc/rc.d/rc.firewall 스크립트에서 "IP Forwarding"을 허용했는지 확인한다. /etc/rc.d/rc.firewall 정책이 제대로 로드되었는지도 확인한다. /etc/rc.d/rc.firewall 스크립트를 수동으로 실행해서 제대로 동작하는지도 확인한다.

여전히 제대로 작동하지 않는다면, 다음 명령의 출력을 잘 확인해 본다.

- "*ifconfig*" : 인터넷 연결이 UP 되어 있는지와 인터넷 연결에 제대로 된 IP 주소가 설정되어 있는지를 확인한다.
- "*netstat -rn*" : 디폴트 게이트웨이(Gateway 부분에 0.0.0.0이 아닌 IP 주소가 있는 것)가 설정되어 있는지 확인한다.
- "*cat /proc/sys/net/ipv4/ip_forward*" : "1"을 출력하면 리눅스 포워딩이 허용되어 있는 것인데 이렇게 나오는지 확인한다.
- 커널 2.0.x에서는 "*/sbin/ipfwadm -F -l*", 커널 2.2.x에서는 "*/sbin/ipchains -F -L*" : 마스커레이딩이 활성화 되어 있는지 확인한다.

- 다섯번째: 마스커레이딩되는 내부의 컴퓨터에서, 인터넷상에 있는 고정 IP 주소로 ping을 해본다 (예를 들면, *ping 152.19.254.81* 와 같이 (이 주소는 LDP 홈페이지인 http://metalab.unc.edu 의 주소이다). 이것이 동작하면, 인터넷을 향한 ICMP 마스커레이딩이 제대로 이루어지고 있다는 것을 의미한다. 만약 동작하지 않으면, 인터넷 연결을 다시 확인한다. 다시 확인했는데도 동작하지 않으면, 예로 든 간단한 rc.firewall 정책을 사용하고 있는지와 커널을 ICMP 마스커레이딩을 포함하여 컴파일했는지 확인한다.
- 여섯번째: 이제 외부의 "IP 주소"로 telnet을 해본다(예를 들면 *telnet 152.2.254.81* (metalab.unc.edu - 이 서버는 부하가 많이 걸리기 때문에 로긴 프롬프트를 받기까지 시간이 걸릴 수도 있다). 어느정도 시간이 지난 후에 로긴 프롬프트를 받는가? 이것이 성공하면, TCP 마스커레이딩이 제대로 동작하고 있는 것이다. 만약 성공하지 못했다면, telnet을 지원하는 것 중에서 여러분이 알고 있는 곳을 시도해 본다. (예를 들면 198.182.196.55 (www.linux.org). 여전히 동작하지 않는다면, 현재 예로 든 간단한 rc.firewall 정책을 사용하고 있는지 확인한다.
- 일곱번째: 이제 외부의 "호스트 이름"으로 telnet을 해본다(예를 들면 *"telnet metalab.unc.edu"* (152.2.254.81). 이것이 성공하면, DNS가 제대로 동작하고 있는 것이다. 이것은 성공하지 못했지만 "네번째" 단계는 성공했다면, 마스커레이딩 내부의 컴퓨터에 DNS 서버관련 설정을 제대로 했는지 확인한다.
- 여덟번째: 마지막 시험으로써, **마스커레이딩 내부의** 컴퓨터에서 웹 브라우져를 통해서 *'INTERNET'*상의 WWW 사이트를 열어보고 브라우져에 표시가 되는지 확인한다. 예를 들어서, `LinuxDocumentationProjectsite`를 접속해 본다. 이것이 성공하면, 모든 것이 훌륭하게 동작하고 있는 것이다!

리눅스 문서 프로젝트의 홈페이지를 볼 수 있다면, **축하한다! 성공했다!** 이 웹 사이트를 제대로 볼 수 있다면, PING, TELNET, SSH등의 표준 네트웍 통신들과, 관련된 IP 마스커레이드 모듈을 적재하면 FTP, Real Audio, IRC DCCs, Quake I/II/III, CuSeeme, VDOLive등도 제대로 동작할 것이다! FTP, IRC, RealAudio, Quake I/II/III등이 제대로 동작하지 않거나 부실하게 동작한다면, "lsmod" 명령으로 관련된 마스커레이딩 모듈들이 제대로 적재되어 있는지 확인하거나 부적절한 서버 포트로 적재되어 있는지 확인한다. 필요한 모듈이 적재되어 있지 않다면, /etc/rc.d/rc.firewall 스크립트가 그 모듈들을 적재하도록 되어 있는지 확인한다. (예를 들면 해당 IP 마스커레이딩 모듈이 있는 부분이 "#" 문자를 제거)

제 6 절 그외의 IP 마스커레이드 관련 사항과 소프트웨어 지원

6.1 IP 마스커레이드의 문제점

어떤 TCP/IP 응용 프로그램들의 프로토콜은, 포트 번호에 대한 것들을 자기 나름대로 가정하거나 그들 데이터의 TCP/IP 주소나 포트 번호를 암호화하기 때문에, 리눅스 IP 마스커레이딩을 통해서는 제대로 동작하지 않는다. 암호화 때문에 문제가 되는 프로토콜을 사용하는 응용프로그램들은, 특정한 프록시 서버라든가, 마스커레이딩 코드 내에 특정 IP 마스커레이드 모듈을 추가해야 작동한다.

6.2 외부로부터 들어오는 서비스

기본적으로, 리눅스 IP 마스커레이딩은 외부로부터 들어오는 서비스들을 전혀 처리하지 못한다. 하지만, 이것을 처리할 수 있는 몇가지 방법이 있다.

만약 높은 수준의 보안을 필요로 하지 않는다면, 간단히 요청이 들어오는 IP 포트를 포워딩해 주면 된다. 이를 수행하는 방법은 여러가지가 있지만, 가장 안정적인 방법은 IPPORTFW를 사용하는 것이다. 더 자세한 정보는 6.7 () 섹션을 참조한다.

외부로부터 들어오는 서비스들에 대해서 어느정도의 인증절차를 갖게 하려면, TCP-wrappers나 Xinetd 등을 사용해서 특정한 IP 주소만 통과시킬 수 있다. 그 도구나 정보를 얻기 위해서는 TIS Firewall Toolkit을 살펴보면 좋을 것이다.

외부로부터 들어오는 서비스의 보안에 대한 자세한 사항은 *TrinityOS* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>` 문서와 *IP Masquerade Resource* `<http://ipmasq.cjb.net>`에서 찾을 수 있을 것이다.

6.3 지원되는 클라이언트 소프트웨어와 그외 설정에 대해 알아둘 점

> **** *Linux Masquerade Application list* `<http://www.tsmservices.com/masq>`에서 리눅스 IP 마스커레이딩을 통해서 동작하는 응용프로그램들에 관한 다량의 우수한 정보들을 얻을 수 있다. 이 사이트는 현재 Steve Grevemeyer가 관리하고 있으며, 그는 풍부한 데이타베이스를 구축해 놓았다. 아주 훌륭한 정보자원이다!**

일반적으로, 표준 TCP와 UDP를 사용하는 응용프로그램들은 잘 작동할 것이다. 만약 어떠한 제안이나 힌트등이 있다면 *IP Masquerade Resource* `<http://ipmasq.cjb.net/>`를 확인해서 자세한 사항을 확인하기 바란다.

6.3.1 IP 마스커레이드와 함께 -동작하는- 네트웍 클라이언트들

일반적인 클라이언트들:

Archie

지원되는 모든 플랫폼, 화일 검색 클라이언트 (모든 archie 클라이언트가 지원되지는 않는다).

FTP

지원되는 모든 플랫폼, active FTP 접속을 위해서 *ip_masq_ftp.o* 커널 모듈 사용.

Gopher client

지원되는 모든 플랫폼.

HTTP

지원되는 모든 플랫폼, 웹 서핑.

IRC

지원되는 여러가지 플랫폼에서 동작하는 모은 IRC 클라이언트, DCC는 *ip_masq_irc.o* 모듈을 통해서 지원.

NNTP (USENET)

지원되는 모든 플랫폼, USENET 뉴스 클라이언트.

PING

모든 플랫폼, ICMP 마스커레이딩 커널 옵션 사용

POP3

지원되는 모든 플랫폼, 이메일 클라이언트.

SSH

지원되는 모든 플랫폼, 보안상 안전한 TELNET/FTP 클라이언트.

SMTP

지원되는 모든 플랫폼, Sendmail, Qmail, PostFix등의 이메일 서버.

TELNET

지원되는 모든 플랫폼, 원격 접속.

TRACEROUTE

UNIX와 Windows 기반 플랫폼, 몇가지 변종들은 동작하지 않을 수 있다.

VRML

Windows(지원되는 모든 플랫폼에서도 어쩌면 가능), 가상 현실.

WAIS client

지원되는 모든 플랫폼.

멀티미디어와 통신 클라이언트:

Alpha Worlds

Windows, 클라이언트-서버 방식의 3D 채팅 프로그램.

CU-SeeMe

지원되는 모든 플랫폼, *ip_masq_cuseeme* 모듈 사용, 자세한 사항은 6.8.2 () 섹션 참조.

ICQ

지원되는 모든 플랫폼. 리눅스 커널이 IPPORTFW를 지원하도록 컴파일되어야 하며, ICQ가 NON-SOCKS 프록시 뒤쪽에서 동작하도록 설정되어야 한다. 이 설정에 대한 자세한 설명은 6.9 () 섹션에서 확인할 수 있다.

Internet Phone 3.2

Windows, Peer-to-peer 방식의 음성 통신, 당신이 통화를 요청하는 경우에는 통신이 가능하지만, 다른 사람들이 당신에게 통화를 요청할 수 있게 하려면 특정 포트를 포워딩하도록 설정해야 한다. 자세한 사항은 6.7 () 섹션을 참조한다.

Internet Wave Player

Windows, 네트웍 스트리밍 오디오(network streaming audio).

Powwow

Windows, Peer-to-peer 방식의 텍스트, 음성, 쪽지 통신, 당신이 통화를 요청하는 경우에는 통신이 가능하지만, 다른 사람들이 당신에게 통화를 요청할 수 있게 하려면 특정 포트를 포워딩하도록 설정해야 한다. 자세한 사항은 6.7 () 섹션을 참조한다.

Real Audio Player

Windows, 네트웍 스트리밍 오디오(network streaming audio), *ip_masq_raudio* UDP 모듈을 사용하면 더 좋은 음질을 얻을 수 있다.

True Speech Player 1.1b

Windows, 네트웍 스트리밍 오디오(network streaming audio)

VDOLive

Windows, *ip_masq_vdolive* 패치 적용.

Worlds Chat 0.9a

Windows, 클라이언트-서버 방식의 3D 채팅 프로그램.

게임 - LooseUDP 패치에 관한 자세한 사항은 6.10 () 섹션을 참조하기 바란다.

Battle.net

동작은 하지만, TCP 포트 116과 118과 UDP 포트 6112를 게임을 실행하는 컴퓨터로 IP 포워딩(IPPORTFW)해줘야 한다. 자세한 사항은 6.7 () 섹션을 참조하기 바란다. FSGS와 Bnetd 서버는 NAT와 잘 동작하도록 만들어져 있지 않기 때문에 IPPORTFW를 필요로 한다.

BattleZone 1.4

LooseUDP 패치를 하고 새로운 NAT와 잘 동작하는 `.DLLsfromActivision`를 사용하면 동작한다.

Dark Reign 1.4

LooseUDP 패치를 하거나 TCP 포트 116과 118과 UDP 포트 6112를 게임을 실행하는 컴퓨터로 IP 포워딩(IPPORTFW)해줘야 한다. 자세한 사항은 6.7 () 섹션을 참조하기 바란다.

Diablo

LooseUDP 패치를 하거나 TCP 포트 116과 118과 UDP 포트 6112를 게임을 실행하는 컴퓨터로 IP 포워딩(IPPORTFW)해줘야 한다. Diablo의 새로운 버젼은 TCP 포트 6112와 UDP 포트 6112만을 사용한다. 자세한 사항은 6.7 () 섹션을 참조하기 바란다.

Heavy Gear 2

LooseUDP 패치를 하거나 TCP 포트 116과 118과 UDP 포트 6112를 게임을 실행하는 컴퓨터로 IP 포워딩(IPPORTFW)해줘야 한다. 자세한 사항은 6.7 () 섹션을 참조하기 바란다.

Quake I/II/III

바로 작동하지만, 마스커레이드 서버 뒤쪽에 Quake I/II/III 플레이어가 두 명 이상 있을 때에는 *ip_masq_quake* 모듈이 필요하다. 또,, 이 모듈은 기본적으로는 Quake I과 QuakeWorld만 지원한다. Quake II를 지원하게 하거나 서버에 기본으로 정해진 이외의 포트로 접속하고자 할 때에는, 3.3 ()와 3.3.1 ()의 모듈 설치 섹션을 참조하라.

StarCraft

LooseUDP 패치를 하고 TCP와 UDP의 6112번 포트를 마스커레이딩 내부의 게임을 실행하려는 컴퓨터로 포워딩(IPPORTFW)해주어야 한다. 자세한 사항은 6.7 () 섹션을 참조하기 바란다.

WorldCraft

LooseUDP 패치를 하면 동작한다.

그외의 클라이언트들:

Linux net-acct package

Linux, 네트웍 관리-어카운팅 관련 패키지

NCSA Telnet 2.3.08

DOS, TELNET, FTP, PING등이 포함된 패키지

PC-anywhere for Windows

MS-Windows, TCP/IP를 통해서 원격으로 PC를 제어한다. 특정한 포트를 포워딩하도록 설정하지 않으면, 클라이언트로는 동작하지만 호스트(서버)로는 동작하지 않는다. 자세한 사항은 6.7 () 섹션을 참조하기 바란다.

Socket Watch

NTP 사용 - 네트웍 시간조절 프로토콜

6.3.2 동작하지 않는 클라이언트:

All H.323 programs

- MS Netmeeting, Intel Internet Phone Beta 2 - 연결은 되지만 목소리는 한쪽으로만(나가는 쪽) 전달된다. 이를 해결 가능한 방법으로써, *Equivalence's PhonePatch* `<http://www.equival.com.au/phonepatch/index.html>` H.323 gateway를 참조하기 바란다.

Intel Streaming Media Viewer Beta 1

서버에 연결할 수 없다.

Netscape CoolTalk

상대편에 연결할 수 없다.

WebPhone

현재는 동작하지 않는다. (이 응용프로그램은 IP 주소에 대한 잘못된 가정을 한다.)

6.4 보안 강도가 보다 높은 IP 방화벽(IPFWADM) 정책

이 섹션에는 커널 2.0.x에서 사용되는 방화벽 도구인 IPFWADM에 대한 더 심도 있는 안내가 실려 있다. IPCHAINS(2.2.x 용)의 정책들에 대해서는 다음 섹션을 참조한다.

이 예는 방화벽/마스커레이드 시스템을 고정 주소를 가진 PPP 접속을 통해서 구축하는 것이다.(동적 PPP에 관한 것도 포함되어 있지만 코멘트 처리되어 있다.) 사용된 인터페이스는 192.168.0.1이고, PPP 인터페이스의 IP 주소는 잘못된 사용을 우려해서 실제와 다른 주소로 대체되었다 :) IP 스푸핑(속임)과 부정적인 라우팅이나 마스커레이딩을 검출하기 위해서 들어오고 나가는 인터페이스를 따로 따로 적었다. 명시적으로 허용되지 않은 것은 **금지되어 있다** (실제적으로는 거부된다). 여기에 나온 rc.firewall 스크립트를 실행하고 나서 IP 마스커레이드 박스가 죽는다면, 여러분의 상황에 맞도록 편집을 했는지를 확인하고, /var/log/messages나 /var/adm/messages의 시스템 로그화일을 검토한다.

PPP, 케이블모뎀 등에 대해서 좀더 이해하기 쉬운, "강도 높은 IP 마스커레이딩의 IPFWADM 정책"에 대해서는, *TrinityOS - Section 10* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>`와 `GreatCircle'sFirewallWWWpage`를 참조하기 바란다.

주의: ISP로부터(PPP, ADSL, 케이블 모뎀 등) 동적 TCP/IP 주소를 받았다면 부팅시에 이 "강도 높은" 정책을 **로드할 수 없다**. 동적으로 IP 주소를 할당받을 때마다 방화벽 정책화일을 다시 로드하거나, /etc/rc.d/rc.firewall 정책화일을 좀더 지능적으로 만들 필요가 있다. PPP 유저의 경우에는, "Dynamic PPP IP fetch" 부분을 주의 깊게 다시 읽어 보고서 적절한 부분의 코멘트를 해제해 준다. 강도 높은 방화벽 정책들과 동적 IP 주소에 관한 더 자세한 사항은 *TrinityOS - Section 10* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>` 문서를 참조한다.

또한 몇가지의 GUI 방식의 방화벽 설정 도구가 있다. 자세한 사항은 7 () 섹션을 참조한다.

마지막으로, 고정 PPP IP 주소를 사용하고 있다면, "ppp_ip = "your.static.PPP.address"" 라고 되어 있는 줄을 여러분의 IP 주소에 맞게 바꾼다.

——————————————————————————-

```
#!/bin/sh
#
# /etc/rc.d/rc.firewall: An example of a semi-STRONG IPFWADM firewall ruleset
#

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

# testing, wait a bit then clear all firewall rules.
# uncomment following lines if you want the firewall to automatically
# disable after 10 minutes.
# (sleep 600; \
# ipfwadm -I -f; \
# ipfwadm -I -p accept; \
# ipfwadm -O -f; \
# ipfwadm -O -p accept; \
# ipfwadm -F -f; \
# ipfwadm -F -p accept; \
# ) &

# Load all required IP MASQ modules
#
#   NOTE:  Only load the IP MASQ modules you need.  All current IP MASQ modules
#          are shown below but are commented from loading.

# Needed to initially load modules
#
/sbin/depmod -a

# Supports the proper masquerading of FTP file transfers using the PORT method
#
/sbin/modprobe ip_masq_ftp

# Supports the masquerading of RealAudio over UDP.  Without this module,
#       RealAudio WILL function but in TCP mode.  This can cause a reduction
#       in sound quality
#
#/sbin/modprobe ip_masq_raudio
```

```
# Supports the masquerading of IRC DCC file transfers
#
#/sbin/modprobe ip_masq_irc


# Supports the masquerading of Quake and QuakeWorld by default.  This modules is
#   for for multiple users behind the Linux MASQ server.  If you are going to play
#   Quake I, II, and III, use the second example.
#
#Quake I / QuakeWorld (ports 26000 and 27000)
#/sbin/modprobe ip_masq_quake
#
#Quake I/II/III / QuakeWorld (ports 26000, 27000, 27910, 27960)
#/sbin/modprobe ip_masq_quake ports=26000,27000,27910,27960


# Supports the masquerading of the CuSeeme video conferencing software
#
#/sbin/modprobe ip_masq_cuseeme

#Supports the masquerading of the VDO-live video conferencing software
#
#/sbin/modprobe ip_masq_vdolive


#CRITICAL:  Enable IP forwarding since it is disabled by default since
#
#           Redhat Users:  you may try changing the options in /etc/sysconfig/network from:
#
#                       FORWARD_IPV4=false
#                             to
#                       FORWARD_IPV4=true
#
echo "1" > /proc/sys/net/ipv4/ip_forward


# Dynamic IP users:
#
#   If you get your IP address dynamically from SLIP, PPP, or DHCP, enable this following
#       option.  This enables dynamic-ip address hacking in IP MASQ, making the life
#       with Diald and similar programs much easier.
#
#echo "1" > /proc/sys/net/ipv4/ip_dynaddr


# Specify your Static IP address here.
#
#   If you have a DYNAMIC IP address, you need to make this ruleset understand your
#   IP address everytime you get a new IP.  To do this, enable the following one-line
#   script.  (Please note that the different single and double quote characters MATTER).
```

```
#
#   You will also need to either create the following link or have your existing
#   /etc/ppp/ip-up script run the /etc/rc.d/rc.firewall script.
#
#       ln -s /etc/rc.d/rc.firewall /etc/ppp/ip-up
#
#   If the /etc/ppp/ip-up file already exists, you should edit it and add a line
#   containing "/etc/rc.d/rc.firewall" near the end of the file.
#
#   If you aren't already aware, the /etc/ppp/ip-up script is always run when a PPP
#   connection comes up.  Because of this, we can make the ruleset go and get the
#   new PPP IP address and update the strong firewall ruleset.
#
#   PPP users:  If your Internet connect is via a PPP connection, the following
                one-line script will work fine.
#
#   DHCP users:  If you get your TCP/IP address via DHCP, you will need to replace
#                the word "ppp0" with the name of your external Internet connection
#               (eth0, eth1, etc).  It should be also noted that DHCP can change
#               IP addresses on you.  To fix this, users should configure their
#               DHCPc or DHCP client to re-run the firewall ruleset when their
#               DHCP lease is renewed.  For DHCPcd users, use the "-c" option.
#
#ppp_ip = "`/sbin/ifconfig ppp0 | grep 'inet addr' | awk '{print $2}' | sed -e 's/.*://'`"
#
ppp_ip = "your.static.PPP.address"


# MASQ timeouts
#
#   2 hrs timeout for TCP session timeouts
#  10 sec timeout for traffic after the TCP/IP "FIN" packet is received
#  60 sec timeout for UDP traffic (MASQ'ed ICQ users must enable a 30sec firewall timeout in ICQ itself
#
/sbin/ipfwadm -M -s 7200 10 60


#############################################################################
# Incoming, flush and set default policy of reject. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
#
/sbin/ipfwadm -I -f
/sbin/ipfwadm -I -p reject

# local interface, local machines, going anywhere is valid
#
/sbin/ipfwadm -I -a accept -V 192.168.0.1 -S 192.168.0.0/24 -D 0.0.0.0/0

# remote interface, claiming to be local machines, IP spoofing, get lost
#
```

```
/sbin/ipfwadm -I -a reject -V $ppp_ip -S 192.168.0.0/24 -D 0.0.0.0/0 -o

# remote interface, any source, going to permanent PPP address is valid
#
/sbin/ipfwadm -I -a accept -V $ppp_ip -S 0.0.0.0/0 -D $ppp_ip/32

# loopback interface is valid.
#
/sbin/ipfwadm -I -a accept -V 127.0.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0

# catch all rule, all other incoming is denied and logged. pity there is no
# log option on the policy but this does the job instead.
#
/sbin/ipfwadm -I -a reject -S 0.0.0.0/0 -D 0.0.0.0/0 -o


###############################################################################
# Outgoing, flush and set default policy of reject. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
#
/sbin/ipfwadm -O -f
/sbin/ipfwadm -O -p reject

# local interface, any source going to local net is valid
#
/sbin/ipfwadm -O -a accept -V 192.168.0.1 -S 0.0.0.0/0 -D 192.168.0.0/24

# outgoing to local net on remote interface, stuffed routing, deny
#
/sbin/ipfwadm -O -a reject -V $ppp_ip -S 0.0.0.0/0 -D 192.168.0.0/24 -o

# outgoing from local net on remote interface, stuffed masquerading, deny
#
/sbin/ipfwadm -O -a reject -V $ppp_ip -S 192.168.0.0/24 -D 0.0.0.0/0 -o

# outgoing from local net on remote interface, stuffed masquerading, deny
#
/sbin/ipfwadm -O -a reject -V $ppp_ip -S 0.0.0.0/0 -D 192.168.0.0/24 -o

# anything else outgoing on remote interface is valid
#
/sbin/ipfwadm -O -a accept -V $ppp_ip -S $ppp_ip /32 -D 0.0.0.0/0

# loopback interface is valid.
#
/sbin/ipfwadm -O -a accept -V 127.0.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0

# catch all rule, all other outgoing is denied and logged. pity there is no
# log option on the policy but this does the job instead.
#
```

```
/sbin/ipfwadm -O -a reject -S 0.0.0.0/0 -D 0.0.0.0/0 -o


#############################################################################
# Forwarding, flush and set default policy of deny. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
#
/sbin/ipfwadm -F -f
/sbin/ipfwadm -F -p deny

# Masquerade from local net on local interface to anywhere.
#
/sbin/ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/24 -D 0.0.0.0/0
#
# catch all rule, all other forwarding is denied and logged. pity there is no
# log option on the policy but this does the job instead.
#
/sbin/ipfwadm -F -a reject -S 0.0.0.0/0 -D 0.0.0.0/0 -o
```

IPFWADM을 사용하면, 여러분은 -I, -O, -F 등의 옵션을 이용해서 특정 사이트로의 접근을 제한할 수 있다. 각 정책 명령들은 위에서부터 아래로 읽혀지고, "-a" 는 기존의 정책에 "덧붙인다"는 것을 유념한다. 그러므로, 특정한 제한 사항은 전반적인 정책보다 앞에 와야 한다. 예를 들면:

-I 를 사용하면, 가장 빠르지만 제한 사항은 내부의 컴퓨터들에게만 적용된다. 방화벽 컴퓨터 자체는 여전히 "금지된" 사이트에 접속할 수 있다. 물론 여러분은 이것들을 조합해서 사용할 수도 있다.

/etc/rc.d/rc.firewall 정책 화일 중에서:

```
... start of -I rules ...

# reject and log local interface, local machines going to 204.50.10.13
#
/sbin/ipfwadm -I -a reject -V 192.168.0.1 -S 192.168.0.0/24 -D 204.50.10.13/32 -o

# local interface, local machines, going anywhere is valid
#
/sbin/ipfwadm -I -a accept -V 192.168.0.1 -S 192.168.0.0/24 -D 0.0.0.0/0

... end of -I rules ...
```

-O 를 사용하면, 패킷들이 마스커레이딩을 먼저 통과하기 때문에 속도는 가장 느리지만, 방화벽 컴퓨터 자체도 금지된 사이트에 접속할 수 없다.

```
... start of -O rules ...

# reject and log outgoing to 204.50.10.13
#
/sbin/ipfwadm -O -a reject -V $ppp_ip -S $ppp_ip/32 -D 204.50.10.13/32 -o

# anything else outgoing on remote interface is valid
#
/sbin/ipfwadm -O -a accept -V $ppp_ip -S $ppp_ip/32 -D 0.0.0.0/0
```

```
... end of -O rules ...
```

-F 를 사용하면, -I 를 사용한 것보다는 조금 더 느리고 역시 마스커레이드 되는 (내부의) 컴퓨터들만 제한하고 방화벽 컴퓨터 자체는 금지된 사이트에 접속할 수 있다.

```
... start of -F rules ...

# Reject and log from local net on PPP interface to 204.50.10.13.
#
/sbin/ipfwadm -F -a reject -W ppp0 -S 192.168.0.0/24 -D 204.50.10.13/32 -o

# Masquerade from local net on local interface to anywhere.
#
/sbin/ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/24 -D 0.0.0.0/0

... end of -F rules ...
```

192.168.0.0/24가 204.50.11.0로 접속할 수 있도록 허용하기 위해 특별한 정책이 필요하지는 않다. 전반적인 정책에 포함되어 있기 때문이다.

위의 정책화일에서 인터페이스를 구성하는 방법은 여러가지가 있을 수 있다. 예를 들어서, "-V 192.168.255.1" 대신에 "-W eth0"라고 적을 수 있고, "-V $ppp_ip" 대신에 "-W ppp0"를 사용할 수 있다. "-V" 를 사용하는 것은 IPCHAINS와의 호환을 위해서 사용된 것이고, IPFWADM만을 사용한다면 전적으로 사용자의 선택이다.

6.5 보안 강도가 보다 높은 IP 방화벽(IPCHAINS) 정책

이 섹션에는 커널 2.2.x에서 사용되는 방화벽 도구인 IPCHAINS에 대한 더 심도 있는 안내가 실려 있다. IPFWADM(2.0.x 용)의 정책들에 대해서는 이전 섹션을 참조한다.

이 예는 방화벽/마스커레이드 시스템을 고정 주소를 가진 PPP 접속을 통해서 구축하는 것이다.(동적 PPP 에 관한 것도 포함되어 있지만 코멘트 처리되어 있다.) 사용된 인>터페이스는 192.168.0.1이고, PPP 인터페이스의 IP 주소는 잘못된 사용을 우려해서 실제와 다른 주소로 대체되었다 :) IP 스푸핑(속임)과 부정적인 라우팅이나 마스커레이딩을 검출하기 위해서 들어오고 나가는 인터페이스를 따로 따로 적었다. 명시적으로 허용되지 않은 것은 **금지되어 있다** (실제적으로는 거부된다). 여기에 나온 rc.firewall 스크립트를 실행하고 나서 IP 마스커레이드 박스가 죽는다면, 여러분의 >상황에 맞도록 편집을 했는지를 확인하고, /var/log/messages나 /var/adm/messages의 시스템 로그화일을 검토한다.

PPP, 케이블모뎀 등에 대해서 좀더 이해하기 쉬운, "강도 높은 IP 마스커레이딩의 IPFWADM 정책"에 대해서는, *TrinityOS - Section 10* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>`와 `GreatCircle'sFirewallWWWpage`를 참조하기 바란다.

주의 #1: 2.2.11보다 버전이 낮은 리눅스 2.2.x 커널은 `IPCHAINSfragmentationbug`를 가지고 있다. 이 때문에, 강도 높은 IPCHAINS 정책을 사용하면 공격에 노출되게 된다. 버그가 수정된 커널로 업그레이드하기 바란다.

주의 #2: ISP로부터(PPP, ADSL, 케이블 모뎀 등) 동적 TCP/IP 주소를 받았다면 부팅시에 이 "강도 높은" 정책을 **로드할 수 없다**. 동적으로 IP 주소를 할당받을 때마다 방화벽 정책화일을 다시 로드하거나, /etc/rc.d/rc.firewall 정책화일을 좀더 지능적으로 만들 필요가 있다. PPP 유저의 경우에는, "Dynamic PPP IP fetch" >부분을 주의 깊게 다시 읽어 보고서 적절한 부분의 코맨트를 해제해 준다. 강도 높은 방화벽 정책들과 동적 IP 주소에 관한 더 자세한 사항은 *TrinityOS - Section 10* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>` >문서를 참조한다.

또한 몇가지의 GUI 방식의 방화벽 설정 도구가 있다. 자세한 사항은 7 () 섹션을 참조한다.

마지막으로, 고정 PPP IP 주소를 사용하고 있다면, "ppp_ip = "your.static.PPP.address"" 라고 되어 있는 줄을 여러분의 IP 주소에 맞게 바꾼다.

————————————————————————————-

```
#!/bin/sh
#
# /etc/rc.d/rc.firewall: An example of a Semi-Strong IPCHAINS firewall ruleset.
#

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

# Load all required IP MASQ modules
#
#   NOTE:  Only load the IP MASQ modules you need.  All current IP MASQ modules
#          are shown below but are commented from loading.

# Needed to initially load modules
#
/sbin/depmod -a

# Supports the proper masquerading of FTP file transfers using the PORT method
#
/sbin/modprobe ip_masq_ftp

# Supports the masquerading of RealAudio over UDP.  Without this module,
#       RealAudio WILL function but in TCP mode.  This can cause a reduction
#       in sound quality
#
/sbin/modprobe ip_masq_raudio

# Supports the masquerading of IRC DCC file transfers
#
#/sbin/modprobe ip_masq_irc


# Supports the masquerading of Quake and QuakeWorld by default.  This modules is
#   for for multiple users behind the Linux MASQ server.  If you are going to play
#   Quake I, II, and III, use the second example.
#
#Quake I / QuakeWorld (ports 26000 and 27000)
#/sbin/modprobe ip_masq_quake
#
#Quake I/II/III / QuakeWorld (ports 26000, 27000, 27910, 27960)
#/sbin/modprobe ip_masq_quake ports=26000,27000,27910,27960


# Supports the masquerading of the CuSeeme video conferencing software
#
```

```
#/sbin/modprobe ip_masq_cuseeme

#Supports the masquerading of the VDO-live video conferencing software
#
#/sbin/modprobe ip_masq_vdolive


#CRITICAL:  Enable IP forwarding since it is disabled by default since
#
#           Redhat Users:  you may try changing the options in /etc/sysconfig/network from:
#
#                       FORWARD_IPV4=false
#                             to
#                       FORWARD_IPV4=true
#
echo "1" > /proc/sys/net/ipv4/ip_forward


# Get the dynamic IP address assigned via DHCP
#
extip="`/sbin/ifconfig eth1 | grep 'inet addr' | awk '{print $2}' | sed -e 's/.*://'`"
extint="eth1"

# Assign the internal IP
intint="eth0"
intnet="192.168.1.0/24"


# MASQ timeouts
#
#   2 hrs timeout for TCP session timeouts
#  10 sec timeout for traffic after the TCP/IP "FIN" packet is received
#  60 sec timeout for UDP traffic (MASQ'ed ICQ users must enable a 30sec firewall timeout in ICQ itself
#
ipchains -M -S 7200 10 60

#############################################################################
# Incoming, flush and set default policy of reject. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
#
ipchains -F input
ipchains -P input REJECT

# local interface, local machines, going anywhere is valid
#
ipchains -A input -i $intint -s $intnet -d 0.0.0.0/0 -j ACCEPT

# remote interface, claiming to be local machines, IP spoofing, get lost
#
ipchains -A input -i $extint -s $intnet -d 0.0.0.0/0 -l -j REJECT
```

```
# remote interface, any source, going to permanent PPP address is valid
#
ipchains -A input -i $extint -s 0.0.0.0/0 -d $extip/32 -j ACCEPT

# loopback interface is valid.
#
ipchains -A input -i lo -s 0.0.0.0/0 -d 0.0.0.0/0 -j ACCEPT

# catch all rule, all other incoming is denied and logged. pity there is no
# log option on the policy but this does the job instead.
#
ipchains -A input -s 0.0.0.0/0 -d 0.0.0.0/0 -l -j REJECT

###############################################################################
# Outgoing, flush and set default policy of reject. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
#
ipchains -F output
ipchains -P output REJECT

# local interface, any source going to local net is valid
#
ipchains -A output -i $intint -s 0.0.0.0/0 -d $intnet -j ACCEPT

# outgoing to local net on remote interface, stuffed routing, deny
#
ipchains -A output -i $extint -s 0.0.0.0/0 -d $intnet -l -j REJECT

# outgoing from local net on remote interface, stuffed masquerading, deny
#
ipchains -A output -i $extint -s $intnet -d 0.0.0.0/0 -l -j REJECT

# anything else outgoing on remote interface is valid
#
ipchains -A output -i $extint -s $extip/32 -d 0.0.0.0/0 -j ACCEPT

# loopback interface is valid.
#
ipchains -A output -i lo -s 0.0.0.0/0 -d 0.0.0.0/0 -j ACCEPT

# catch all rule, all other outgoing is denied and logged. pity there is no
# log option on the policy but this does the job instead.
#
ipchains -A output -s 0.0.0.0/0 -d 0.0.0.0/0 -l -j REJECT

###############################################################################
# Forwarding, flush and set default policy of deny. Actually the default policy
# is irrelevant because there is a catch all rule with deny and log.
#
```

```
ipchains -F forward
ipchains -P forward DENY

# Masquerade from local net on local interface to anywhere.
#
ipchains -A forward -i $extint -s $intnet -d 0.0.0.0/0 -j MASQ
#
# catch all rule, all other forwarding is denied and logged. pity there is no
# log option on the policy but this does the job instead.
#
ipchains -A forward -s 0.0.0.0/0 -d 0.0.0.0/0 -l -j REJECT
```

IPCHAINS를 사용하면, 여러분은 ”input”, ”output”, ”forward” 규칙을 통해서 특정 사이트와의 통신을 제한할 수 있다. 각 정책 명령들은 위에서부터 아래로 읽혀지고, ”-A” 는 기존의 정책에 ”덧붙인다”는 것을 유념한다. 그러므로, 특정한 제한 사항은 전반적인 정책보다 앞에 와야 한다. 예를 들면:

”input” 규칙: 가장 빠르지만 제한은 내부의 컴퓨터들에게만 적용된다. 방화벽 컴퓨터 자체는 여전히 ”금지된” 사이트에 접속할 수 있다. 물론 여러분은 이것들을 조합해서 사용할 수도 있다.

/etc/rc.d/rc.firewall 정책 화일 중에서:

```
... start of "input" rules ...

# reject and log local interface, local machines going to 204.50.10.13
#
/sbin/ipfwadm -I -a reject -V 192.168.0.1 -S 192.168.0.0/24 -D 204.50.10.13/32 -o

# local interface, local machines, going anywhere is valid
#
/sbin/ipfwadm -I -a accept -V 192.168.0.1 -S 192.168.0.0/24 -D 0.0.0.0/0

... end of "input" rules ...
```

”output”을 사용하면, 패킷들이 마스커레이딩을 먼저 통과하기 때문에 속도는 가장 느리지만, 방화벽 컴퓨터 자체도 금지된 사이트에 접속할 수 없다.

```
... start of "output" rules ...

# reject and log outgoing to 204.50.10.13
#
/sbin/ipfwadm -O -a reject -V $ppp_ip -S $ppp_ip/32 -D 204.50.10.13/32 -o

# anything else outgoing on remote interface is valid
#
/sbin/ipfwadm -O -a accept -V $ppp_ip -S $ppp_ip/32 -D 0.0.0.0/0

... end of "output" rules ...
```

”forward”를 사용하면, ”input”을 사용한 것보다는 조금 더 느리고 역시 마스커레이드 되는 (내부의) 컴퓨터들만 제한하고 방화벽 컴퓨터 자체는 금지된 사이트에 접속할 수 있다.

```
... start of "forward" rules ...
```

```
# Reject and log from local net on PPP interface to 204.50.10.13.
#
/sbin/ipfwadm -F -a reject -W ppp0 -S 192.168.0.0/24 -D 204.50.10.13/32 -o

# Masquerade from local net on local interface to anywhere.
#
/sbin/ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/24 -D 0.0.0.0/0

... end of "forward" rules ...
```

192.168.0.0/24가 204.50.11.0로 접속할 수 있도록 허용하기 위해 특별한 정책이 필요하지는 않다. 전반적인 정책에 포함되어 있기 때문이다.

IPFWADM와는 다르게, 위의 정책화일에서 인터페이스를 구성하는 방법은 오직 한가지 뿐이다. IPCHAINS는 "-i eth0" 옵션을 사용한다. "-V"는 IPFWADM의 방법으로 IPCHAINS와의 호환을 위해서 사용된 것이고, IPFWADM만을 사용한다면 전적으로 사용자의 선택이다.

6.6 여러개의 내부 네트웍을 IP 마스커레이딩하는 법

여러개의 내부 네트웍을 마스커레이딩하는 것이 매우 간단하다. 우선 내부와 외부의 모든 네트웍이 제대로 동작하는지 확인해야 한다. 그런 후에 네트웍 트래픽이 내부의 다른 컴퓨터들에게도 전달되고 인터넷으로 마스커레이딩되도록 만들어야 한다.

다음으로, 내부의 인터페이스에 마스커레이딩을 사용가능하도록 해줘야 한다. 이 예는 eth1 (192.168.0.1)와 eth2 (192.168.1.1)의 두개의 내부 인터페이스가 외부로 향하는 eth0 인터페이스로 마스커레이딩되도록 설정하는 것이다. rc.firewall 정책화일에 다음 내용을 추가한다:

- IPFWADM을 사용하는 2.0.x 커널용

```
#Enable internal interfaces to communication between each other
/sbin/ipfwadm -F -a accept -V 192.168.0.1 -D 192.168.1.0/24
/sbin/ipfwadm -F -a accept -V 192.168.1.1 -D 192.168.0.0/24

#Enable internal interfaces to MASQ out to the Internet
/sbin/ipfwadm -F -a masq -W eth0 -S 192.168.0.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -F -a masq -W eth0 -S 192.168.1.0/24 -D 0.0.0.0/0
```

- IPCHAINS를 사용하는 2.2.x 커널용

```
#Enable internal interfaces to communication between each other
/sbin/ipchains -A forward -i eth1 -d 192.168.1.0/24
/sbin/ipchains -A forward -i eth2 -d 192.168.0.0/24

#Enable internal interfaces to MASQ out to the Internet
/sbin/ipchains -A forward -j MASQ -i eth0 -s 192.168.0.0/24 -d 0.0.0.0/0
/sbin/ipchains -A forward -j MASQ -i eth0 -s 192.168.1.0/24 -d 0.0.0.0/0
```

6.7 IP 마스커레이드와 전화 접속

1. 인터넷에 연결하기 위해서 자동으로 전화 접속을 하도록 설정하고자 한다면, *Diald*를 사용한 전화걸기나 *PPPd*의 새 버전을 사용하는 것이 좋을 것이다. 구성이 더 나은 Diald를 사용하는 것을 권장한다.

2. Diald를 설정하기 위해서는, *Setting Up Diald for Linux Page* <http://home.pacific.net.sg/harish/diald.config.html>나 *TrinityOS - Section 23* <http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>를 살펴보기 바란다.
3. 일단 Diald와 IP 마스커레이딩이 제대로 설정되고 나면, 마스커레이드되는 클라이언트들이 웹이나 telnet, ftp등의 접속을 하려고 하면 리눅스 box가 자동으로 인터넷 연결을 할 것이다.
4. 처음 접속 때는 시간 초과가 있을 수도 있는데, 아날로그 모뎀을 사용한다면 어쩔 수 없다. 모뎀 자체의 접속과 PPP 접속을 위한 시간 때문에, 클라이언트 프로그램들(웹 브라우저 등)이 시간 초과를 일으킬 수 있다. 하지만 늘 그런 것은 아니다. 만약 이런 현상이 일어나면, 단지 재시도(이를테면, 웹 페이지를 다시 보기)를 하면 그 다음부터는 잘 동작할 것이다. 또는, *echo "1" > /proc/sys/net/ipv4/ip_dynaddr*라고 커널에 옵션을 주어서 이러한 초기 설정에 관한 것을 해결할 수도 있다.

6.8 IPPORTFW, IPMASQADM, IPAUTOFW, REDIR, UDPRED 등과 기타의 포트 포워딩 도구들

IPPORTFW, IPAUTOFW, REDIR, UDPRED 등과 기타 다른 프로그램들은 리눅스 IP 마스커레이드를 위한 일반적은 TCP또는 UDP 포트 포워딩 도구들이다. 이러한 도구들은 일반적으로, 현재의 FTP, Quake 등을 위한 특정한 IP 마스커레이드 모듈들과 함께 사용되거나 대체해서 사용된다. 포트 포워딩 도구들을 사용하면, 인터넷으로부터 오는 접속들을, IP 마스커레이딩 뒤에서 내부 주소만 가지고 있는 컴퓨터로 전달해 줄 수 있다. 이러한 포워딩 기능은 TELNET, WWW, SMTP, FTP (특별한 패치를 필요로 한다 - 아래를 볼 것), ICQ 와 다른 많은 네트웍 프로토콜들을 처리할 수 있다.

주의: 만약 IP 마스커레이딩 없이 단지 포트 포워딩만을 하길 원한다 해도, **여전히** 커널과 IPFWADM나 IPCHAINS 정책 내에 IP 마스커레이딩 기능을 추가하고서 리눅스의 포트 포워딩 도구들을 사용해야 한다.

그러면 이것들의 차이는 무엇인가? IPAUTOFW, REDIR와 UDPRED(모든 URL들은 2.5 () 섹션에 있다) 등은 IP 마스커레이드 사용자가 이 기능을 사용하기 위해서 필요한 초기의 도구들이었다. 시간이 흐르고, 리눅스 IP 마스커레이드가 발전하면서, 이 도구들은 더 지능적인 해결책인 IPPORTFW로 대체되었다. 새로운 도구들을 사용가능하게 되었기 때문에, IPQUTOFW와 REDIR와 같은 예전의 도구들을 사용하는 것은 *매우 바람직하지 않다*. 이들은 커널과 함께 제대로 동작하지 못하거나 심지어 여러분의 리눅스 서버를 **파괴**할 수도 있다.

2.0.x 버젼의 IPPORTFW나 2.2.x 버젼의 IPMASQADM을 IPPORTFW와 함께 사용하기 전에, 다른 포트 포워딩 도구들을 사용하면 네트웍 보안 문제를 일으킬 수 있다. 그 이유는 이러한 도구들은 TCP/UDP 포트를 포워딩하기 위해서 기본적으로 패킷 방화벽에 구멍을 만들기 때문이다. 이것이 리눅스 머신에 위협을 주지는 않지만, 패킷이 포워딩되는 내부의 컴퓨터에게는 문제가 될 수도 있다. 큰 문제는 아니지만, IPPORTFW의 제작자인 Steven Clarke은 다음과 같이 말한다:

```
"해당하는 IPFWADM/IPCHAINS 정책에 들어맞도록, 포트 포워딩은
마스커레이딩 함수에서만 불려진다. 마스커레이딩은 IP 포워딩으로
확장된다. 그래서, ipportfw는 입력과 ipfwadm 정책 모두에 들어맞는
패킷만을 볼 수 있다."
```

이러한 이유로, 강력한 방화벽 정책을 사용하는 것이 중요하다. 강력한 방화벽 정책에 대한 자세한 것은 6.4 ()과 6.5 () 섹션을 참조하기 바란다.

이제, IPPORTFW 포워딩을 2.0.x나 2.2.x 커널에 사용하기 위해서는, 리눅스 커널이 IPPORTFW를 지원하도록 재컴파일해야 한다.

- 2.0.x 커널 사용자들은 아래와 같은 간단한 커널 옵션 패치를 해야 한다.
- 2.2.x 커널 사용자들은 IPMASQADM을 통해서 이미 IPPORTFW 커널 옵션을 사용할 수 있을 것이다.

6.8.1 2.0.x 커널에서 IPPORTFW 사용

우선, /usr/src/linux 디렉토리에 가장 최신의 2.0.x 커널이 존재하는지 확인한다. 만약 없다면, 3.1 () 섹션에서 자세한 사항을 참조한다. 다음으로, 2.5 () 섹션에서 "ipportfw.c" 프로그램과 "subs-patch-x.gz" 커널 패치를 다운로드해서 /usr/src/ 디렉토리에 압축을 푼다.

주의: "subs-patch-x.gz"의 화일명에서 "x"는 그 사이트에서 구할 수 있는 가장 최신의 버젼으로 대치한다.

이제, IPPORTFW 패치(subs-patch-x.gz)를 리눅스 디렉토리로 복사한다.

```
cp /usr/src/subs-patch-1.37.gz /usr/src/linux
```

다음에, IPPORTFW 커널 옵션을 생성하기 위해서 커널 패치를 한다:

```
cd /usr/src/linux
zcat subs-patch-1.3x.gz | patch -p1
```

다음으로, FTP 접속을 내부의 서버로 포트 포워딩하고자 한다면, 2.5 () 섹션에서 구할 수 있는 **새로운** *IP_MASQ_FTP* 모듈 패치를 해야 한다. 자세한 사항은 이 섹션의 나중 부분에 나와 있다.

자, 3.1 () 섹션에 있는 것처럼 커널을 컴파일할 차례이다. 커널을 설정하는 단계에서 IPPORTFW 옵션에 YES라고 하도록 한다. 일단 컴파일이 끝나고 새로운 커널로 리부트하고 나면, 다시 이 섹션으로 돌아온다.

이제 새로 컴파일한 커널을 사용해서, 실제의 "IPPORTFW" 프로그램을 컴파일하고 설치한다.

```
cd /usr/src
gcc ipportfw.c -o ipportfw
mv ipportfw /usr/local/sbin
```

이제, 예를 들어서 모든 웹 접속(포트 80)을 마스커레이드되는 내부의 머신 중에서 192.168.0.10을 주소로 같은 머신으로 포워드하려 한다고 하자.

주의: 일단 포트 80을 포트 포워딩하면, 리눅스 IP 마스커레이드 서버는 그 포트를 더이상 사용하지 못한다. 더 구체적으로, 만약 여러분이 마스커레이드 서버에 이미 웹 서버를 운영하고 있고 포트 80을 내부의 마스커레이드되는 컴퓨터로 포트 포워딩한다면, 모든 인터넷 사용자들은 IP 마스커레이드 서버에서 보내는 페이지가 아닌 -내부의- 웹 서버에서 보내는 페이지를 볼 것이다. 이를 해결하기 위한 유일한 방법은 예를 들어 8080과 같은 다른 포트를 내부의 마스커레이드되는 머신으로 포트 포워딩하는 것이다. 이렇게 하면 되긴 하지만, 모든 인터넷 사용자들은 내부의 마스커레이드되는 웹 서버에 접속하기 위해서 URL에 *:8080*을 덧붙여야 한다.

어쨌든, 포트 포워딩을 사용하기 위해서는, */etc/rc.d/rc.firewall* 정책화일을 편집한다. 다음에 있는 내용을 추가하되, "$extip"는 여러분의 인터넷 IP 주소로 대치한다.

주의: 만약 여러분이 ISP 로부터(PPP, ADSL, 케이블 모뎀, 기타 등등) 동적 TCP/IP 주소를 받아서 사용한다면, /etc/rc.d/rc.firewall 정책화일을 더 지능적으로 만들어야 할 것이다. 강력한 정책들과 동적 IP 주소에 관한 자세한 사항은 *TrinityOS - Section 10* <http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>을 참조하기 바란다.

```
/etc/rc.d/rc.firewall
--

#echo "Enabling IPPORTFW Redirection on the external LAN.."
#
/usr/local/sbin/ipportfw -C
/usr/local/sbin/ipportfw -A -t$extip/80 -R 192.168.0.10/80

--
```

자 이제 됐다! /etc/rc.d/rc.firewall 정책화일을 다시 실행시키고 시험해 보자!

만약에 "ipfwadm: setsockopt failed: Protocol not available" 라는 에러 메시지를 보게 된다면, 새로 컴파일한 커널을 사용하고 있지 않는 것이다. 새로운 커널을 제대로 된 위치에 옮겨 놓았는지, LILO를 재 실행했는지 확인하고 다시 한번 리부트한다.

FTP 서버의 포트 포워딩:

FTP를 내부의 머신으로 포트 포워딩하고자 한다면, 일이 좀 더 복잡해진다. 그 이유는 표준의 *IP_MASQ_FTP* 커널 모듈이 이러한 목적으로 만들어져 있지 않기 때문이다. 다행히 Fred Viles가 이러한 목적으로 동작하도록 수정된 IP_MASQ_FTP 모듈을 작성했다. 정확히 무엇이 문제인지 알고 싶다면, Fred가 문서를 매우 잘 작성해 놓았으니 그것을 다운로드해 보기 바란다. 이 패치는 다소 실험적인 면이 있다는 것도 알아두기 바란다. 또한 현재 이 패치는 2.0.x 커널용에서만 사용할 수 있다는 것도 알아두기 바란다. 2.2.x 커널로의 포팅도 어느 정도 이루어져 있지만, 여기에 도움을 주고 싶다면 `FredViles-fv@episupport.com`로 직접 이메일을 보내기 바란다.

이제 다음 과정을 거쳐서 2.0.x 패치를 한다:

- 우선 이 섹션의 앞 부분에 있는 것과 같이 IPPORTFW 커널 패치를 가한다.
- 2.5 () 섹션에 수록된 Fred Viles의 FTP 서버에서 "msqsrv-patch-36"를 다운로드하고 /usr/src/linux에 넣는다.
- "cat msqsrv-patch-36 | patch -p1"라고 명령해서 이 새로운 코드로 커널을 패치한다.
- 이제, 원래의 *"ip_masq_ftp.c"* 커널 모듈을 새로운 것으로 대체한다.
  - mv /usr/src/linux/net/ipv4/ip_masq_ftp.c /usr/src/linux/net/ipv4/ip_masq_ftp.c.orig
  - mv /usr/src/linux/ip_masq_ftp.c /usr/src/linux/net/ipv4/ip_masq_ftp.c
- 마지막으로 새로운 코드가 적용된 커널을 생성해서 인스톨한다.

다 됐으면, /etc/rc.d/rc.firewall 정책화일을 편집해서 다음 내용을 추가하되 "$extip"는 여러분의 인터넷 IP 주소로 대치한다.

주의: 만약 여러분이 ISP 로부터(PPP, ADSL, 케이블 모뎀, 기타 등등) 동적 TCP/IP 주소를 받아서 사용한다면, /etc/rc.d/rc.firewall 정책화일을 더 지능적으로 만들어야 할 것이다. 강력한 정책들과 동적 IP 주소에 관한 자세한 사항은 *TrinityOS - Section 10* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>`을 참조하기 바란다.

이 예는 위에서와 같이 모든 FTP 접속(포트 21)을 내부의 마스커레이드되는 머신 중 192.168.0.10의 주소를 갖는 머신으로 포트 포워딩할 것이다.

주의: 일단 포트 21을 포트 포워딩하면, 리눅스 마스커레이드 서버는 그 포트를 더 이상 사용하지 못할 것이다. 더 구체적으로, 만약 리눅스 마스커레이드 서버에 이미 FTP 서버를 운영하고 있다면, 모든 인터넷 사용자의 FTP 접속은 IP 마스커레이드 서버가 아니라 -내부의- FTP 서버로 갈 것이다.

```
/etc/rc.d/rc.firewall
--

#echo "Enabling IPPORTFW Redirection on the external LAN.."
#
/usr/local/sbin/ipportfw -C
/usr/local/sbin/ipportfw -A -t$extip/21 -R 192.168.0.10/21

--
```

자 이제 됐다! /etc/rc.d/rc.firewall 정책화일을 다시 실행시키고 시험해 보자!

만약에 "ipchains: setsockopt failed: Protocol not available" 라는 에러 메시지를 보게 된다면, 새로 컴파일한 커널을 사용하고 있지 않는 것이다. 새로운 커널을 제대로 된 위치에 옮겨 놓았는지, LILO를 재 실행했는지 확인하고 다시 한번 리부트한다. 새로운 커널을 사용하고 있는 것이 확실하다면, "ls /proc/net"이라고 명령해서 "ip_portfw" 화일이 있는지 확인한다. 없다면, 커널을 설정하는 단계에서 무엇인가 빠트렸을 것이다. 커널을 다시 만든다.

6.8.2 2.2.x 커널에서 IPPORTFW와 함께 IPMASQADM 사용

우선, /usr/src/linux 디렉토리에 최신의 2.2.x 커널이 있는지 확인한다. 없다면, 3.1 () 섹션에서 자세한 사항을 참조한다. 다음으로, 2.6 () 섹션에서 "ipmasqadm.c" 프로그램을 다운로드해서 /usr/src/ 디렉토리에 넣는다.

다음으로, 3.1 () 섹션에 있는 것과 같이 2.2.x 커널을 컴파일해야 한다. 커널을 설정하는 단계에서 IPPORTFW 옵션에 YES 라고 한다. 일단 커널을 컴파일해서 리부트한 후에 이 섹션으로 돌아온다.

이제, IPMASQADM 도구를 컴파일하고 설치한다:

```
cd /usr/src
tar xzvf ipmasqadm-x.tgz
cd ipmasqadm-x
make
make install
```

이제, 예를 들어서 모든 웹 접속(포트 80)을 내부의 마스커레이드되는 머신중에서 192.168.0.10을 주소로 갖는 머신으로 포트 포워딩한다고 하자.

주의: FTP 접속을 포트 포워딩하기 위해서 수정된 *IP_MASQ_FTP* 모듈이 현재로는 2.2.x 커널에서 동작하지 않을 지도 모른다. 하지만 이를 시험해 보고 싶다면, 이 모듈을 2.2.x 커널용으로 포팅해 보라. 그리고 Ambrose 와 David에게 여러분의 결과물을 메일로 보내 주기 바란다.

주의: 일단 포트 80을 포트 포워딩하고 나면, 리눅스 IP 마스커레이드 서버는 그 포트를 사용하지 못한다. 더 구체적으로, 만약 마스커레이드 서버에 이미 웹 서버를 운영하고 있다면, 모든 인터넷 사용자들은 여러분의 IP 마스커레이드 서버가 아닌 -내부의- 웹 서버에서 웹 페이지를 받아 볼 것이다.

어쨌든, 포트 포워딩을 하기 위해서는 /etc/rc.d/rc.firewall 정책 화일을 편집한다. 다음의 내용을 추가하되, "$extip"를 여러분의 인터넷 IP 주소로 대치한다.

주의: 만약 ISP로부터(PPP, ADSL, 케이블 모뎀, 기타 등등) 동적 TCP/IP 주소를 받아서 사용하고 있다면, /etc/rc.d/rc.firewall 정책화일을 좀더 지능적으로 만들 필요가 있다. 강력한 방화벽 정책과 동적 IP 주소에 관한 자세한 사항에 대해서는 *TrinityOS - Section 10* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>`을 참조하기 바란다. 여기에 힌트를 한가지 제공한다: PPP 사용자들을 위한 /etc/ppp/ip-up 화일.

```
/etc/rc.d/rc.firewall
--

#echo "Enabling IPPORTFW Redirection on the external LAN.."
#
/usr/sbin/ipmasqadm portfw -f
/usr/sbin/ipmasqadm portfw -a -P tcp -L $extip 80 -R 192.168.0.10 80

--
```

자 이제 됐다! /etc/rc.d/rc.firewall 정책화일을 다시 실행시키고 시험해 보자!

만약에 "ipchains: setsockopt failed: Protocol not available" 라는 에러 메시지를 보게 된다면, 새로 컴파일한 커널을 사용하고 있지 않는 것이다. 새로운 커널을 제대로 된 위치에 옮겨 놓았는지, LILO를 재 실행했는지 확인하고 다시 한번 리부트한다. 새로운 커널을 사용하고 있는 것이 확실하다면, "ls /proc/net/ip_masq"라고 명령해서 "portfw" 화일이 있는지 확인한다. 없다면, 커널을 설정하는 단계에서 무엇인가 빠트렸을 것이다. 커널을 다시 만든다.

6.9 CU-SeeMe와 리눅스 IP 마스커레이드

리눅스 IP 마스커레이드는 *"ip_masq_cuseeme"* 커널 모듈을 통해서 CuSeeme를 지원한다. 이 커널 모듈은 /etc/rc.d/rc.firewall 스크립트에서 메모리에 적재되어야 한다. 일단 "ip_masq_cuseeme" 모듈일 설치되면, 원격의 reflector들이나 사용자들에게 접속 신호를 보내거나 접속을 받아들일 수 있게 된다.

주의: CuSeeme를 사용하기 위해서는 예전의 IPAUTOFW 도구 대신에 IPPORTFW 도구를 사용할 것을 권장한다.

CuSeeme를 설정하는 데 있어서 더 확실한 정보가 필요하다면, *Michael Owings's CuSeeMe page* `<http://www.swampgas.com/vc/ipmcus.htm>`에서 미니-하우투를 보거나 `TheIPMasqueradeResources`에서 미니-하우투의 미러 페이지를 볼 수 있을 것이다.

6.10 Mirabilis ICQ

리눅스 마스커레이드 서버의 뒤에서 ICQ를 사용하는 방법은 두가지가 있다. 한가지 방법은 새로운 ICQ 마스커레이드 모듈을 사용하는 것이고, 다른 한가지는 IPPORTFW를 사용하는 것이다.

ICQ 모듈은 몇가지 이득과 함께 제한도 있다. 이 모듈은 간단한 설정으로 마스커레이드 서버 뒤에서 여러 명이 ICQ를 사용할 수 있게 한다. ICQ 클라이언트에서 특별한 설정을 필요로 하지도 않는다. 그러나, 현재는 화일 전송과 실시간 채팅이 되지 않는다.

IPPORTFW를 설정해서 사용하면, 마스커레이드 서버와 ICQ 클라이언트 모두에 몇가지 설정을 변경시켜 줘야 하지만, ICQ의 메시지 기능, URL 기능, 채팅, 화일 전송 등 모든 것이 동작할 것이다.

Andrew Deryabin의 `djsf@usa.net` 2.2.x 커널을 위한 ICQ IP 마스커레이드 모듈에 관심이 있다면, 2.6 () 섹션에서 자세한 사항을 확인하기 바란다.

마스커레이드 서버 뒤에서 ICQ를 사용하기 위해 다소 고전적인 방법을 쓰길 원한다면 다음과 같이 한다:

- 우선, 리눅스 커널에 IPPORTFW 기능을 포함시킨다. 자세한 사항은 6.7 () 섹션을 참조한다.
  - 다음으로, 다음의 내용을 /etc/rc.d/rc.firewall 화일에 추가한다. 이 예는 외부로 통하는 여러분의 인터넷 IP 주소를 10.1.2.3으로, 내부의 마스커레이드되는 ICQ 클라이언트를 192.168.0.10으로 가정했다: IPFWADM을 사용하는 2.0.x 커널의 예:

```
두 가지 예를 포함시켰다: 어떤 것이든 잘 동작할 것이다:

예 #1
--
/usr/local/sbin/ipportfw -A -t10.1.2.3/2000 -R 192.168.0.10/2000
/usr/local/sbin/ipportfw -A -t10.1.2.3/2001 -R 192.168.0.10/2001
/usr/local/sbin/ipportfw -A -t10.1.2.3/2002 -R 192.168.0.10/2002
/usr/local/sbin/ipportfw -A -t10.1.2.3/2003 -R 192.168.0.10/2003
/usr/local/sbin/ipportfw -A -t10.1.2.3/2004 -R 192.168.0.10/2004
/usr/local/sbin/ipportfw -A -t10.1.2.3/2005 -R 192.168.0.10/2005
/usr/local/sbin/ipportfw -A -t10.1.2.3/2006 -R 192.168.0.10/2006
/usr/local/sbin/ipportfw -A -t10.1.2.3/2007 -R 192.168.0.10/2007
```

```
/usr/local/sbin/ipportfw -A -t10.1.2.3/2008 -R 192.168.0.10/2008
/usr/local/sbin/ipportfw -A -t10.1.2.3/2009 -R 192.168.0.10/2009
/usr/local/sbin/ipportfw -A -t10.1.2.3/2010 -R 192.168.0.10/2010
/usr/local/sbin/ipportfw -A -t10.1.2.3/2011 -R 192.168.0.10/2011
/usr/local/sbin/ipportfw -A -t10.1.2.3/2012 -R 192.168.0.10/2012
/usr/local/sbin/ipportfw -A -t10.1.2.3/2013 -R 192.168.0.10/2013
/usr/local/sbin/ipportfw -A -t10.1.2.3/2014 -R 192.168.0.10/2014
/usr/local/sbin/ipportfw -A -t10.1.2.3/2015 -R 192.168.0.10/2015
/usr/local/sbin/ipportfw -A -t10.1.2.3/2016 -R 192.168.0.10/2016
/usr/local/sbin/ipportfw -A -t10.1.2.3/2017 -R 192.168.0.10/2017
/usr/local/sbin/ipportfw -A -t10.1.2.3/2018 -R 192.168.0.10/2018
/usr/local/sbin/ipportfw -A -t10.1.2.3/2019 -R 192.168.0.10/2019
/usr/local/sbin/ipportfw -A -t10.1.2.3/2020 -R 192.168.0.10/2020
--

예 #2
--
port=2000
while [ $port -lt 2020 ]
  do
      /usr/local/sbin/ipportfw -A t10.1.2.3/$port -R 192.168.0.10/$port
      port=$((port+1)
  done
--
```

IPCHAINS를 사용하는 2.2.x 커널의 예:

```
두 가지 예를 포함시켰다: 어떤 것이든 잘 동작할 것이다:

예 #1
--
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2000 -R 192.168.0.10 2000
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2001 -R 192.168.0.10 2001
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2002 -R 192.168.0.10 2002
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2003 -R 192.168.0.10 2003
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2004 -R 192.168.0.10 2004
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2005 -R 192.168.0.10 2005
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2006 -R 192.168.0.10 2006
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2007 -R 192.168.0.10 2007
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2008 -R 192.168.0.10 2008
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2009 -R 192.168.0.10 2009
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2010 -R 192.168.0.10 2010
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2011 -R 192.168.0.10 2011
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2012 -R 192.168.0.10 2012
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2013 -R 192.168.0.10 2013
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2014 -R 192.168.0.10 2014
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2015 -R 192.168.0.10 2015
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2016 -R 192.168.0.10 2016
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2017 -R 192.168.0.10 2017
```

```
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2018 -R 192.168.0.10 2018
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2019 -R 192.168.0.10 2019
/usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 2020 -R 192.168.0.10 2020
--

예 #2
--
port=2000
while [ $port -lt 2020 ]
  do
      /usr/local/sbin/ipmasqadm portfw -a -P tcp -L 10.1.2.3 $port -R 192.168.0.10 $port
      port=$((port+1)
  done
--
```

  – 새로운 rc.firewall이 준비되면, 간단히 "/etc/rc.d/rc.firewall"라고 명령해서 정책을 다시 로드하고 제대로 되는지 확인한다. 만약 에러가 난다면, 커널에 IPPORTFW 지원을 포함시키지 않았거나 rc.firewall 화일에서 오타가 났을 것이다.
  – 이제, ICQ의 Preferences(설정)–>Connection(접속)에서, "Behind a LAN(LAN을 통해서 접속)"과 "Behind a firewall or Proxy(방화벽이나 프록시를 통해서 접속)"을 설정한다. 이제, "Firewall Settings(방화벽 설정)"을 클릭하고 "I don't use a SOCK5 proxy(SOCK5 프록시를 사용하지 않음)"로 설정한다. 예전에는 ICQ의 "Firewall session timeouts(방화벽 접속 제한시간)"을 "30"초로 하는 것을 권장했지만, ICQ가 불안정해짐이 알려졌다. stock timeout setting을 선택하고 단순히 마스커레이드 서버의 제한시간을 160초로 변경하면 ICQ가 더 안정적이 된다는 것이 알려졌다. 이 제한시간을 변경하는 법은 3.3 ()과 3.3.1 () 정책에서 확인할 수 있다. 마지막으로, Next를 클릭하고 "Use the following TCP listen ports..(다음의 TCP 포트를 확인..)"부분을 "2000"에서 "2020"으로 설정한다. 이제 "완료"를 클릭한다. 이제 ICQ가 변경사항을 반영하기 위해 ICQ를 재시작할 것을 요구할 것이다. 사실은, 필자는 모든 것이 제대로 되도록 하기 위해서는 Windows9x를 리부트해야만 했지만 다른 사람들은 다르게 말한다. 그러니 안전하게 하려면 두가지를 모두 해 본다(ICQ 재시작, 리부트)

- 아울러 알리고 싶은 것은, 어떤 사용자는 단순히 포트 4000을 그의 ICQ 클라이언트로 포트 포워딩하는 것이 가장 잘 동작한다고 말했다. 그는 ICQ를 기본설정에서 변경하지 않고도 모든 것이(채팅, 화일 전송, 기타 등등) 잘 동작했다고 했다. 이 문제는 여러분이 선택할 문제이지만, 이러한 의견에 대해서도 알고 싶어하는 사람이 있을 것이다.

6.11 게임: LooseUDP 패치

LooseUDP 패치는, 일반적으로 리눅스 IP 마스커레이드 서버 뒤에서 동작하면서 UDP 접속을 사용하는, NAT와 잘 동작하는 게임들을 할 수 있도록 해 준다. 현재, LooseUDP는 2.0.36이상의 커널에 패치로서 제공되고 2.2.3이상의 커널에는 이미 자체 포함되어 있다. 이를 사용하기 위해서는, 몇가지 일만 해주면 된다:

- /usr/src/linux 디렉토리에 최신의 2.0.x 커널 소스가 들어 있는지 확인한다.
- 버전 2.0.x에서는 절대적으로 필요한 것: 2.5 () 섹션에서 IPPORTFW 패치를 다운로드 하고 이 하우투의 6.7 () 섹션에 설명된 대로 설치한다.
- 2.5 () 섹션에서 LooseUDP 패치를 다운로드한다.

이제, LooseUDP 패치를 /usr/src/linux 디렉토리에 넣는다. 그 다음에 다음과 같이 명령한다:

```
압축된 패치 화일일 때: zcat loose-udp-2.0.36.patch.gz | patch -p1

압축되지 않는 패치 화일일 때: cat loose-udp-2.0.36.patch | patch -p1
```

이제, "patch" 프로그램의 버전에 따라서, 다음과 같은 메시지를 볼 것이다:

```
patching file `CREDITS'
patching file `Documentation/Configure.help'
patching file `include/net/ip_masq.h'
patching file `net/ipv4/Config.in'
patching file `net/ipv4/ip_masq.c'
```

패치의 제일 처음에서만 "Hunk FAILED"라는 메시지를 보게 된다면, 심각한 문제는 아니다. 아마도 오래된 패치 화일일 테지만 동작할 것이다. 하지만 만약 패치가 완전히 실패한다면, IPPORTFW 커널 패치를 "먼저" 적용했는지 확인해 본다.

패치가 설치되면, 3.1 () 섹션에 나와 있는 대로 커널을 재구성하고 "IP: loose UDP port managing (EXPERIMENTAL) (CONFIG_IP_MASQ_LOOSE_UDP) [Y/n/?]" 옵션에서 "Y"라고 한다.

일단 LooseUDP 기능이 추가된 새 커널을 사용하면, NAT와 잘 동작하는 게임들은 잘 동작할 것이다. BattleZone이나 다른 게임들을 NAT와 잘 동작하도록 해 주는 패치들을 구할 수 있는 몇가지 URL들이 있다. 자세한 사항은 6.3.1 () 섹션을 참조한다.

제 7 절 자주 묻는 질문들(FAQ)

유용한 FAQ가 있다면, `ambrose@writeme.com`과 `dranch@trinnet.net`으로 보내주기 바란다. 질문을 명확하게 표시하고 적절한 답변을 달아주기 바란다. 미리 감사드린다!

7.1 IP 마스커레이드를 바로 사용할 수 있는 리눅스 배포본은 어떤 것입니까?

여러분의 리눅스 배포본이 IP 마스커레이드를 바로 사용할 수 없다 해도 걱정하지 마십시요. 단지 이 하우투에 나온 대로 커널을 재컴파일하기만 하면 됩니다.

주의: 이 표를 완벽히 채우는데 도움을 주고자 할 때에는 `ambrose@writeme.com`이나 `dranch@trinnet.net`으로 이메일을 주십시요.

- Caldera < v1.2 : NO - ?
- Caldera v1.3 : YES - 2.0.35 기반
- Caldera v2.2 : YES - 2.2.5 기반
- Debian v1.3 : NO - ?
- Debian v2.0 : NO - ?
- Debian v2.1 : NO - 2.2.1 기반
- DLX Linux v? : ? - ?
- DOS Linux v? : ? - ?
- Hal91 Linux v? : ? - ?
- Linux Mandrake v5.3 : YES - ?

- Linux Mandrake v6.0 : YES - 2.2.5 기반
- Linux PPC vR4 : NO - ?
- Linux Pro v? : ? - ?
- LinuxWare v? : ? - ?
- MkLinux v? : ? - ?
- MuLinux v3rl : YES - ?
- Redhat < v4.x : NO - ?
- Redhat v5.0 : YES - ?
- Redhat v5.1 : YES - ?
- Redhat v5.2 : YES - 2.0.36 기반
- Redhat v6.0 : YES - 2.2.5 기반
- Slackware v3.0 : ? - ?
- Slackware v3.1 : ? - ?
- Slackware v3.2 : ? - ?
- Slackware v3.3 : ? - 2.0.34 기반
- Slackware v3.4 : ? - ?
- Slackware v3.5 : ? - ?
- Slackware v3.6 : ? - ?
- Slackware v3.9 : ? - 2.0.37pre10 기반
- Slackware v4.0 : ? - ?
- Stampede Linux v? : ? - ?
- SuSE v5.2 : YES - ?
- SuSE v5.3 : YES - ?
- SuSE v6.0 : YES - ?
- SuSE v6.1 : YES - 2.2.5 기반
- Tomsrbt Linux v? : ? - ?
- TriLinux v? : ? - ?
- TurboLinux v? : ? - ?
- Yggdrasil Linux v? : ? - ?

7.2 IP 마스커레이드가 동작하기 위한 최소한의 하드웨어 사양과 제한사항은 무엇입니까? 성능은 어느정도입니까?

16MB RAM을 갖는 486/66으로도 1.54Mb/s T1을 100% 처리하고도 남았었다! 마스커레이드는 386SX-16s에서 8BM RAM을 가지고서도 잘 동작한다고 알려져 있다. 그러나, 마스커레이드 항목이 500개가 넘으면 리눅스 IP 마스커레이드도 버벅이기 시작한다는 것도 알아두어야 할 것이다.

리눅스 IP 마스커레이드를 잠시나마 멈추게 할 수 있는 유일한 응용프로그램으로는, 필자가 아는 한 GameSpy뿐이다. 그 이유는 목록을 갱신할 때, 매우 짧은 시간동안 10,000개의 빠른 접속을 필요로 하기 때문이다. 이 일이 끝날 때까지는, 마스커레이드 테이블이 "꽉" 차게 된다. 자세한 사항은 FAQ의 **??** () 섹션을 살펴보기 바란다.

말하는 김에 몇가지 더:

TCP와 UDP에는 4096개의 동시 접속 한계가 있다. 이 한계는 */usr/src/linux/net/ipv4/ip_masq.h*에서 값을 건드리면 수정될 수 있다 - 위쪽 한계인 32000 정도도 괜찮다. 한계치를 수정하고 싶다면 - PORT_MASQ_BEGIN 와 PORT_MASQ_END 값을 32K보다 높고 64K보다 낮은 범위로 수정하면 된다.

7.3 모든 설정을 확인했지만, 여전히 IP 마스커레이드가 동작하지 않습니다. 어떻게 해야 합니까?

- 우선 마음을 가라 앉히십시요. 차를 한잔하든가, 커피나, 음료수라든가. 그리고 좀 쉬십시요. 일단 마음이 진정되었으면, 아래에 있는 제안대로 따라하십시요. 리눅스 IP 마스커레이드를 설정하는 것은 어렵지 않지만, 몇가지 생소한 개념이 있을 수 있습니다.
- 자, 이제 다시 5 () 섹션에 있는 것대로 따라하십시요. 마스커레이드를 처음 사용하는 사람들 중에서 문제가 발생한 경우의 99%는 그 섹션을 보지 않았을 겁니다.
- *IP Masquerade Mailing List Archives* `<http://www.indyramp.com/lists/masq/>`를 확인해 보십시요. 여러분의 질문이나 문제들 중 대부분은 보통의 질문들이고, 간단히 Archive를 검색해 보면 답을 찾을 수 있을 겁니다.
- *TrinityOS* `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS.wri>` 문서를 확인해 보십시요. 그 문서는 2.0.x 와 2.2.x 커널에서 IP 마스커레이딩을 사용하는 것에 대해서 다루고 있으며, PPPd, DialD, DHCP, DNS, Sendmail이나 그외의 주제들을 다루고 있습니다.
- 혹시나 여러분이 ROUTED나 GATED를 실행하고 있지는 않는지 확인하십시요. 확인하기 위해서는, "ps aux | grep -e routed -e gated"라고 명령해 보십시요.
- 여러분의 질문을 IP 마스커레이드 메일링 리스트로 보내십시요(자세한 것은 FAQ의 다음 섹션을 참조하십시요). 단, 질문에 대한 답을 IP 마스커레이딩 Archive에서 찾을 없을 때만 보내십시요. 이메일을 보낼 때는 5 () 섹션에 있는 대로 실행했을 때의 결과를 반드시 포함시키십시요!!
- 여러분의 질문을 관련된 리눅스 NNTP 뉴스그룹으로 보내십시요.
- `ambrose@writeme.com`과 `dranch@trinnet.net`으로 이메일을 보내십시요. 하지만, 우리들에게 질문하는 것보다 IP 마스커레이딩 메일링 리스트에서 원하는 답을 얻기가 쉬울 겁니다.
- 여러분의 설정을 다시 확인하십시요. :-)

7.4 IP 마스커레이드나 IP 마스커레이드 개발자 메일링 리스트에 참가하거나 보기 위해서는 어떻게 해야 합니까?

리눅스 IP 마스커레이딩 메일링 리스트에 참가하는 방법에는 두가지가 있습니다. 첫번째 방법은 `masq-request@indyramp.com`으로 메일을 보내는 것입니다. 리눅스 IP 마스커레이딩 개발자 메일링 리스

트에 참가하기 위해서는, masq-dev-request@indyramp.com으로 메일을 보내십시요. 더 자세한 사항은 아래의 기사를 참조하십시요.

- 이메일을 통해서 가입(참가)하기: 메일 내용이나 제목에 "subscribe"라는 단어를 입력하십시요. 마스커레이드 주 메일링 리스트나 마스커레이드 개발자 리스트의 축약판에만 가입하기 원한다면 (그 리스트로 보내지는 모든 이메일이 일주일에 한번 한 개의 "큰" 이메일로 여러분에게 보내질 겁니다), 메일 내용이나 제목에 "subscribe" 대신 "subscribe digest"라고 입력하십시요. 일단 서버가 여러분의 요청을 받으면, 여러분이 요청한 리스트에 가입시키고 여러분에게 패스워드를 보낼 겁니다. 이 패스워드를 어딘가에 저장해 놓으십시요. 옵션을 변경하거나 리스트에서 탈퇴할 때 필요합니다.

두번째 방법은 웹 브라우져를 이용해서 가입하는 겁니다. 마스커레이드 주 리스트에 가입하려면 http://www.indyramp.com/masq-list/의 형식에 맞춰서 가입하고, 마스커레이드 개발자 리스트에 가입하려면 http://www.indyramp.com/masq-dev-list/를 이용하십시요.

일단 가입되고 나면, 가입된 리스트에서 이메일을 받을 겁니다. 또 한가지 알려둘 것은 리스트에 가입하든 가입하지 않든, 두 리스트의 archive를 볼 수 있습니다. 자세한 방법은 위에 있는 두 개의 웹 URL을 참조하십시요.

마지막으로 알려둘 것은, 마스커레이드 리스트에 글을 올리기 위해서는 처음에 가입했던 계정과 주소를 이용해야 한다는 겁니다.

메일링 리스트나 메일링 리스트 archive에 관련한 문제가 발생하면, RobertNovak에게 연락하십시요.

7.5 IP 마스커레이드가 프록시(Proxy)나 NAT 서비스와 다른 점은 무엇입니까?

```
Proxy:  프록시 서버는 다음 환경에서 사용가능: Win95, NT, Linux, Solaris, 기타.

             장점:   + 한개의 IP 주소 ; 저렴함
                     + 더 나은 성능(웹 등)을 위해서 선택적으로 캐쉬 사용

             단점:   - 프록시 서버 뒤에 있는 모든 응용프로그램들이
                       프록시 서비스(SOCKS)를 지원해야 하고 프록시 서버를
                       사용하도록 설정되어야 한다
                     - 웹 카운터나 웹 통계 프로그램을 혼란시킨다

    프록시 서버는, IP 마스커레이드와 같이, 단 한개의 공식적인 IP 주소를
    사용하고, 내부 LAN에 있는 클라이언트들(웹 브라우저 등등)에게 번역자
    역할을 한다. 이 프록시 서버는 내부 네트웍으로부터 오는 TELNET, FTP,
    웹과 같은 접속을 한 개의 인터페이스를 통해서 받아들인다. 그리고 나서,
    프록시 서버 자체에서 접속을 요청한 것처럼 바꾸어서 외부로 보낸다.
    일단 원격의 인터넷 서버가 요청한 정보를 보내오면, 프록시 서버는
    TCP/IP 주소를 내부의 클라이언트의 주소로 다시 변경하고 내부에서
    요청했던 호스트로 보내준다. 이러한 것을 "프록시(대리인)" 서버라고
    부른다.

             주의 :  내부의 머신들에서 사용하는 모든 응용프로그램은
                     *반드시* 프록시 서버 사용을 지원해야 한다. 예를 들면,
                     Netscape나 몇몇 좋은 TELNET이나 FTP 클라이언트들.
                     프록시 서버를 지원하지 않는 클라이언트들은 동작하지
                     않을 것이다.
```