가정용 네트워크를 위한 인터넷 게이트웨이로서 레드햇 리눅스 6.X 활용

Paul Ramsey <pramsey@refractions.net> 2000년 6월 22일

레드햇 6 및 관련된 다른 종류의 리눅스를 소규모 가정용 또는 사무실 네트워크에 인터넷 게이트웨이로 운영하도록 구성하는 데 대한 간단한 튜토리얼. 다루어진 주제는 매스커레이딩(masquerading), DNS, DHCP, 기초 수준의 보안을 포함한다.

목 차

제 1 절 소개 **2**

1.1 버전 . . . 2

1.2 저작권 . . . 3

제 2 절 꽂기 **3**

2.1 허브를 가진 경우 . . . 3

2.2 허브가 없는 경우 . . . 3

2.3 네트워크 카드가 하나만 있는 경우 . . . 4

제 3 절 네트워킹 구성하기 **4**

3.1 네트워크 드라이버를 구성하기 . . . 5

3.1.1 두 개의 동일한 네트워크 카드 . . . 6

3.2 내부 네트워크를 구성하기 . . . 6

3.2.1 네트워크 장치 . . . 6

3.2.2 DHCP 서버 . . . 7

3.2.3 클라이언트 컴퓨터 . . . 8

3.2.4 DNS 서버 . . . 9

3.2.5 내부 네트워크를 시험하기 . . . 10

3.3 외부 네트워크를 구성하기 . . . 10

3.3.1 정적인 IP를 가진 경우 . . . 10

3.3.2 DHCP를 사용하는 경우 . . . 11

3.3.3 기발함과 예외들 . . . 11

3.3.4 네트워크 엔트리를 보기 . . . 12

3.4 보안 . . . 13

제 4 절 매스커레이딩을 구성하기 **14**

제 5 절 Problems 문제점 **15**

5.1 ICQ가 작동하지 않는다 15

5.2 레드햇 6.X가 아닌 칼데라 2.X를 가지고 있다. 15

5.3 내부 장비 중의 한 대가 웹 서버가 되기를 바란다 15

제 1 절 소개

이 페이지는 레드햇 6.X를 가정용 네트워크 또는 소규모 사무용 네트워크를 위한 인터넷 게이트웨이로 설정하기 위한 간단한 설명서를 포함한다. 지시는 매우 간략화 되었는데, 어떠한 특정 경우도 논의하지 않을 것이고, 어떤 네트워크 주소가 사용되는지에 대한 몇 가지 가정을 제시할 것이다. 매우 중요한 가정은 다음과 같다:

- 여러분은 전용선(fulltime Cable) 또는 ADSL로 인터넷에 연결되어 있다.
- 여러분은 자신의 컴퓨터들 중 적어도 한 대에 `6.X`를 성공적으로 설치할 수 있다. 이런 안내는 또한 다양한 상표로 맥밀란 출판사(MacMillan Publishing)에 의해 배포되는 `Mandrake6.X`와 같은 레드햇 파생품에도 유효하다는 것을 알아 두자.
- 여러분의 리눅스 컴퓨터는 내부에 설치된 네트워크 카드를 두 개 가지며 둘 다 리눅스와 호완된다.
- 만일 여러분이 한 대 이상의 컴퓨터를 네트워크로 연결한다면 이더넷 허브를 가져야 한다. 아니면, 단 한 대의 컴퓨터를 연결한다면 크로스-오버 케이블(cross-over cable)을 가져야 한다.
- 여러분은 리눅스 장비에서 텍스트 파일을 편집하는 방법을 안다.
- 여러분은 `root`로서 장비에 로그인 할 수 있다. 여러분은 리눅스 CDROM으로부터 RPM 패키지를 설치하는 방법을 안다.

만일 여러분이 이러한 가정의 어느 하나라도 만족시키지 않으면 이 문서는 아마도 여러분한테 소용이 없을 것이다.

설치 과정 동안 해야 하는 특별한 것이 있는 것은 아니다. 여러분이 쉽게 이해하고 성취 가능한 지시를 단순히 선택하라. 이 문서는 무(無)에서부터 네트워킹을 처리하기 위해 필요한 모든 것을 설치하는 길잡이 역할을 한다. 설치하는 동안 설치하거나 구성해야 하는 것에 대해 어떤 가정도 전제로 하지 않는다. 그러한 것들이 작동함을 보장하고 무슨 정보가 어디로 가는지에 대해 혼동하지 않도록 모든 구성은 레드햇과 함께 재공되는 GUI 구성 도구를 사용하기 보다는 시스템 구성 파일을 직접 편집하게 될 것이다. 한편으로는, 이것이 좀 더 힘들 것이다. 또 한편으로는, (가령, X가 작동하지 않는, 또는 헤드리스 서버(headless server)를 설정하는 경우에) 여러분은 상이한 배포본과 상황에 훨씬 더 수월하게 전이할 수 있는 지식을 얻게 될 것이다.

1.1 버전

이 문서의 최종 버전은 HTML 버전용으로 `http://www.coastnet.com/\~{}pramsey/linux/homenet.html`, SGML 버전용으로 `http://www.coastnet.com/\~{}pramsey/linux/homenet.sgml`에서 언제나 구할 수 있다.

- 1999년 12월 21일: 최초 버전
- 2000년 1월 2일: John Mellor로부터 외부 네트워크 개선에 관한 제안을 취합
- 2000년 1월 22일: Chris Lea로부터 동일한 네트워크 카드와 IP 알리어싱(aliasing)에 관한 정보에 대한 사소한 업데이트

- 2000년 3월 16일: Nelson Gibbs로부터 네임서버 보안과 칼데라를 지원하는 데 관한 정보
- 2000년 6월 22일: 레드햇 6.2 구성의 개선된 점을 문서화. Kerr First로부터 더 많은 PPPoE 정보를 얻음

1.2 저작권

저작권 ©2000, Paul Ramsey.

이 문서는, 다음과 같은 제한 사항을 적용하여, 무료로, 전체로 또는 부분적으로 재생산될 수 있다:

- 상기한 저작권 고지와 이 허가 고지는 반드시 모든 완전하거나 부분적인 복사본에 완전하게 유지되어야 한다.
- 어떠한 번역 또는 파생된 저작물이라도 배포 전에 작성되는 동안 반드시 저자에 의해 인가되어야 한다.
- 만일 이 저작물을 부분으로 배포하려면, 이 매뉴얼의 완전한 버전을 구하는 방법을 위한 지시를 반드시 포함시켜야 하고, 완전한 버전을 구하기 위한 수단을 제공하여야 한다.
- 만일 적절한 인용이 주어진다면 이 허가 고지가 없더라도 작은 부분은 다른 저작물에서 검토나 인용을 위한 예증으로서 재생산될 수 있다.

이러한 규정에 대한 예외가 학문적인 목적을 위해서 인정된다: 저자한테 편지를 써서 요청하라. 이러한 제한 사항은, 학생과 교육자인 여러분을 제한하는 것은 아니라, 우리를 저자로서 보호하고자 함이다.

제 2 절 꽂기

여러분이 허브를 사용하고 있는지 아닌지에 따라서 네트워크 토폴로지(network topology)는 조금이라도 다를 것이다. 저자는 오로지 RJ45 케이블 연결을 사용한 네트워킹을 다룰 것이다. 동축 케이블(thin coax)을 다루지 않는다. 동축 케이블을 사용하면 허브를 갖추지 않고서도 다수의 컴퓨터를 네트워크를 구성할 수 있다. 하지만, 연결 끝을 마무리하는 등과 같이 더욱 조심스러워야 한다. 여러분이 이미 네트워킹을 알고 있다면 이들 지시는 대개가 중복되는 것일 것이다.

2.1 허브를 가진 경우

만일 여러분이 허브를 가지고 있다면, 여러분의 네트워크는 *이것* <http://www.coastnet.com/~pramsey/linux/w_hub.gif>과 같이 보일 것이다.

리눅스 박스에 있는 `eth0` 카드를 서비스 제공자에 의해 공급된 케이블(또는 그 구성에서 작동하는 것으로 여러분이 아는 것)을 사용하여 케이블 모뎀 또는 ADSL 박스에 연결하라. 어떤 때에는 케이블 모뎀이 크로스오버로 연결되기도 쉽고, 때로는 케이블을 통해 직통(straight-through)으로 연결되기도 하기 때문에 이는 중요하다. 회사가 제공하는 것이 여러분이 사용하기 원하는 것이다.

리눅스 박스에 있는 `eth1` 카드를 직통선(straight-through cable)을 사용하여 허브에 연결하라. 직통선을 사용하여 나머지 컴퓨터 전부를 허브에 연결하라.

2.2 허브가 없는 경우

만일 여러분이 허브를 가지지 않았다면, 크로스오버 케이블을 사용하여, 컴퓨터 한 대를 여전히 리눅스 박스에 연결할 수 있다. 여러분의 토폴로지는 *이것* <http://www.coastnet.com/~pramsey/linux/wo_hub.gif>과 같이 보일 것이다.

리눅스 박스에 있는 eth0 카드를 서비스 제공자에 의해 공급된 케이블을 사용하여 케이블 모뎀 또는 ADSL 박스에 연결하라. 리눅스 박스에 있는 eth1 카드를 크로스오버 케이블을 사용하여 다른 컴퓨터에 연결하라.

2.3 네트워크 카드가 하나만 있는 경우

이는 바람직한 구성은 아니다. (이 구성에서는 여러분의 내부 및 외부 네트워크이 동일한 물리적 네트워크에 있고, 때문에 이론적으로는 크래킹을 쉽게 허용하게 된다. 실제로, 위험은 아마도 매우 낮다.) 하지만, 받아들여질 수 있다. 여러분의 이정은 가지각색일지 모른다.

리눅스 커널은 "IP 알리어싱" 지원을 포함한다. 이는 하나의 이더넷 카드가 동시에 두 개의 상이한 IP 주소를 다루는 것을 허용한다. 레드햇과 맨드레이크와 함께 선적된 표준적인 커널은 기본적으로 IP 알리어싱을 위한 지원을 포함한다. 오로지 한 장의 이더넷 카드만으로 게이트웨이를 설정하기 위해서는 다음에 보이는 모든 코드 예제에서 간단히 eth1를 eth0:0로 치환하라.

단일 카드 상황에서, DHCP 서버를 구동하는 것은 바람직하지 않다.

모든 장비와 케이블 모뎀 또는 ADSL 박스를 허브에 꽂아라. 행운을 빌고, 계속하라.

제 3 절 네트워킹 구성하기

자, 지금까지 여러분은 게이트웨이 컴퓨터에 리눅스를 설치했다. 여러분은 네트워크 카드 중 하나를 구성했고, 인터넷에 접속을 설정했다. 그렇지만, 우리는 무에서부터 시작할 것이고 어떤 것도 전혀 구성되지 않았다고 가정할 것이다.

root로서 로그인 하라. 이 문서에 나온 모든 지시는 여러분이 root로서 로그인 했다고 가정한다.

리눅스 커널은 두 장의 이더넷 카드를 eth0와 eth1로서 참조한다. 이제부터는 또한 그것들에 대해 언급하게 될 방법이 될 것이다. 애로 사항은, 어느 것이 어떤 것인지이다. 경우의 적어도 50%는 쓰이도록 보증한다면, 판별하는 "간단한" 방법이 있다. 마더보드를 가진 컴퓨터를 책상 위에 수평적으로 놓고 뒷 패널(back panel)을 여러분을 향하게 한다. (만일 여러분이 그걸 열고 그 위에서 무엇인가 하게 되는 것과 같이 하도록 한다.) 가장 왼편의 카드는 eth0이다. – 여러분은 약간의 마스킹 테이프로 라벨을 붙이고자 할지 모른다. 지금, 종이 위에 eth0와 eth1 둘 모두의 제조와 모델을 써 두어라.

자, eth0와 eth1이 커널에 의해 자동적으로 인식되는지 여부를 보자. ifconfig eth0와 ifconfig eth1를 입력한다. 두 경우에 있어, 만일 커널이 카드를 인식하고 있으면, 여러분은 (숫자와 그 밖의 것은 상이할 수 있음을 명심하여) 이와 같은 것을 보아야 한다:

```
eth0   Link encap: Ethernet   HWaddr 00:60:67:4A:02:0A
       inet addr:0.0.0.0  Bcast:0.0.0.0  Mask:255.255.255.255
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:466 errors:0 dropped:0 overruns:0 frame:0
       TX packets:448 errors:0 dropped:0 overruns:0 carrier:0
       collisions:85 txqueuelen:100
       Interrupt:10 Base address:0xe400
```

만일 커널이 네트워크 카드를 인식하지 못한다면 이와 같은 것을 보게 될 것이다:

```
eth0: error fetching interface information: Device not found.
```

3.1 네트워크 드라이버를 구성하기

만일 카드를 둘 다 찾았으면 다음 섹션으로 넘어 가라. 아니면, 이 섹션을 읽어라.

자, 이미 아는 바와 같이 카드가 하나 혹은 둘 모두가 커널에 의해 인식되지 않는다. 실제로, 이것은 문제가 되지 않는다. 반드시 해야 할 것은 카드를 찾는 방법을 커널에 좀 더 명시적으로 알리는 것이다. 여기에는 우여곡절이 많고, 그 모두를 다루지는 않을 것이다. 현행의 것이 어려워질 때 `EthernetHOWTO`로 바꾸어라. 여기에 몇 가지 요약된 조언이 있다.

- *여러분은 PCI 네트워크 카드를 가지고 있다.* 어떤 드라이버도 존재하지 않을 만큼 최신의 것은 아니라고 가정한다면, 여러분은 아마도 성공할 것이다. 여러분은 `/proc/pci`를 통해 해독하고 제조원과 모델을 적어 둠으로써 네트워크 카드(와 다른 것들)에 대해 많은 것을 종종 알수 있다.
- *여러분은 ISA 네트워크 카드를 가지고 있다.* 여러분은 카드가 작동하는 IO 기본 주소와 IRQ를 꼭 알아 낼 수 있어야 할 것이다. 여러분은 매뉴얼을 가지고 있다, 맞는가? 맞는가? 만일 그렇지 않으면, 이번이 제작사의 웹 사이트를 찾아내고 어떤 온라인 참고 자료가 있는지 아는 좋은 기회가 될 것이다. 아니면, 만일 오래된 DOS 구성 디스켓을 가지고 있으면, DOS로 부팅하고 주소와 IRQ를 판독하고 설정하는 설정 프로그램(setup program)이 있는지 알아 내라.
- *여러분은 ISA 플러그 앤 플레이 카드를 가지고 있다.* 우선 여러분은 카드를 구성하는 방법을 배워야 할 것이다. – `Plug'n'PlayHOWTO`를 읽어라. 다행스럽게도, 일단 여러분이 카드를 구성하게 되면 IO 기본 주소와 IRQ가 무엇인지 정확하게 알게 될 것이다.

이제, 여러분은 `eth0`와 `eth1`의 제조원과 모델이 무엇인지 알기 때문에 `EthernetHOWTO`의 `compatibilitypage`로 옮겨가서 카드를 찾아낼 수 있다. 추천된 드라이버와 카드가 필요로 할지 모르는 특정한 옵션에 대한 정보를 메모하라. 적어 두라.

구성 파일을 편집할 때다! 편집하게 될 파일은 `/etc/conf.modules`이다. 여러분이 선택한 텍스트 에디터(text editor)로 이 파일을 열어라. 이 파일에 포함될 수 있는 매우 많은 가능성과 조합이 있기 때문에 한 예로서 나 자신의 게이트웨이를 제공할 것이다. 나는 VIA Rhine 칩에 기반한 PCI 10/100Mb 카드와 10Mb NE2000 ISA 복제품(clone)을 가진다. 나는 내부 네트워크를 위해 100Mb 카드를, 외부 연결을 위해 10Mb 카드를 사용한다. 나의 `/etc/conf.modules` 파일은 이것과 같다:

```
alias parport_lowlevel parport_pc
alias eth0 ne
options ne io=0x300 irq=10
alias eth1 via-rhine
```

나의 `conf.modules` 파일을 다음과 같이 펼쳐 보인다.

- 첫 번째 행은 프린팅을 위한 병렬 포트를 구성하는 것이다. 아마 여러분도 유사한 행을 가질 것이다. 그대로 내버려 두라.
- 두 번째 행(`alias eth0 ne`)은 `eth0` 장치를 위한 ne 드라이버를 사용하도록 커널에 알린다.
- 세 번째 행(`options ne io=0x300 irq=10`)은 ne 드라이버가 ISA 카드를 발견하게 될 IO 주소와 IRQ 인터럽트를 알린다. 만일 여러분이 ISA 카드를 가지고 있다면, 아마 이런 종류의 지시를 사용하여야 할 것이다. 여러분의 카드를 위한 정확한 정보를 사용하여 드라이버, IO와 IRQ 지시자를 교체하기만 하라.
- 네 번째 행(`alias eth1 via-rhine`)은 `eth1`를 위해 via-rhine 드라이버를 사용하도록 커널한테 알린다. 나의 `eth1` 카드가 PCI 카드이기 때문에 IO 또는 IRQ 정보를 제공할 필요가 없다: PCI 부시스템은 장치를 자동적으로 구성한다.

여러분은 두 개의 카드를 위해 conf.modules에 alias 기재사항을 가진다는 것을 보장하고, 모든 ISA 카드를 위한 options 행을 정확하게 만들고자 할 것이다. 여러분은 설치하는 동안 여러분이 구성한 이더넷 카드를 위한 conf.modules에 이미 여러 행들을 이미 가지고 있을지 모른다.

여러분이 conf.modules을 편집하는 것을 완료하였을 때, ifconfig eth0와 ifconfig eth1을 다시 시도하라. 만일 제작사의 매뉴얼이 없이 여러분이 IO 주소와 IRQ를 실수하고 있다면 시행착오를 여러 번 겪을 것이다.

3.1.1 두 개의 동일한 네트워크 카드

그래서, 여러분은 진짜로 진짜로 현명하였고, 여러분의 리눅스 게이트웨이를 위해 동일한 네트워크 카드를 두 개 샀다. 여러분이 그것들을 함께 작동하도록 할 수 없다면? 걱정하지 마라. 그것들이 공존하도록 하는 것은 /etc/conf.modules에서 정확한 구문을 사용하는 것의 문제일 뿐이다. 이 예를 위해 주소와 IRQ 번호는 만들어졌고, 조화된 한 쌍의 NE2000 복제품을 구매했다고 가정하겠다. /etc/conf.modules 파일은 이것과 같다:

```
alias eth0 ne
alias eth1 ne
options ne io=0x330,0x360 irq=7,9
```

The addressing options are all given on the same line, and the first number for each addressing type is for eth0, the second number for eth1. 주소를 지정하는 옵션은 동일한 행에 모두 주어지고, 각 주소를 지정하는 형태를 위한 첫 번째의 수는 eth0를 위한 것이고, 두 번째의 수는 eth1를 위한 것이다.

3.2 내부 네트워크를 구성하기

"내부 네트워크"는 여러분의 모든 가정/사무실 장비가 통하게 되는 네트워크이다. "외부 네트워크"는 리눅스 박스의 다른 쪽에 있는 대규모의 무시무시한 인터넷이다. 대체로, 내부 네트워크는 리눅스 박스에 의해 외부 네트워크로부터 완전히 격리될 것이다. 이는 중간 정도의 강력한 방화벽으로서 운용될 것이다.

3.2.1 네트워크 장치

여러분의 드라이버가 작동하고 있고 ifconfig에서 eth0와 eth1를 둘 다 볼 수 있으므로 내부의 가정용 네트워크를 설정하여야 하는 때이다. eth1에 내부 네트워크를, eth0에 외부 장치를 둔다고 가정하겠다.

여러분의 내부 네트워크는 개인용 네트워크가 될 것이고, 이를 위해서 내부 네트워크 연결을 위해 예약된 특정한 네트워크에 있게 될 것이다: 192.168.1.0. 만약 친구를 감명시키기를 원하면, 이는 "개인용 C 클래스"이다.

우선 네트워킹이 가능한 상태인지 확인할 필요가 있다. /etc/sysconfig/network파일을 편집하여 다음의 행들이 있는지 확인하라.

```
NETWORKING=yes
FORWARD_IPV4=yes
```

첫 번째 행은 부트(boot) 시에 네트워크 장치가 불려 오도록 리눅스한테 알린다. 두 번째 행은 IP 전달(IP forwarding)을 가능케 하도록 리눅스한테 알린다. 섹션 4에서 매스커레이딩을 구성하는 것을 시작할 때 이것이 필요하다.

래드햇 6.2 주의: IP 전송과 매스커레이딩을 적절하게 지원하기 위해서 레드햇 6.2는 /etc/sysctl.conf 파일에 변경을 필요로 한다. 다음의 행들이 있는지 확인하고 정확한 값을 지정한다:

```
net.ipv4.ip_forward = 1
net.ipv4.ip_always_defrag = 1
```

레드햇과 레드햇 파생품을 위한 모든 네트워크 인터페이스 설정은 /etc/sysconfig/network-scripts 디렉토리 내의 파일들에 포함된다. 그 디렉토리에 들어가서 새 파일로 ifcfg-eth1를 만든다. ifcfg-eth1 파일 내에 다음과 같은 내용을 둔다:

```
DEVICE=eth1
IPADDR=192.168.1.1
ONBOOT=yes
```

이 코드는 부트 시에 eth1을 구성하고 개별적인 IP 주소를 주도록 네트워크 스크립트한테 알린다. 다음과 같은 명령어와 함께 새 설정을 사용하여 여러분의 네트워크를 활성화 하라: /etc/rc.d/init.d/network restart

3.2.2 DHCP 서버

DHCP 서버는 IP 주소를 가지는 내부 가정용 네트워크 상의 장치를 자동적으로 구성할 것이다. 이것은 랩톱을 가진 사람들을 위해 매우 유용하다. 그들이 자신들의 장비를 간단히 꼽기만 하면 즉시 적절하게 구성될 수 있다. 만일 여러분이 내부 네트워크 상에서 DHCP 서버를 원하지 않는다면 다음 섹션으로 넘어 가라.

우선 여러분은 DHCP 서버가 설치되었는지 확인할 필요가 있다. 리눅스 CD를 마운트하고 dhcp RPM을 설치하라. 이제 /etc/dhcpd.conf 파일을 편집하여 다음과 같은 내용을 넣는다:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.2 192.168.1.60;
  default-lease-time 86400;
  max-lease-time 86400;
  option routers 192.168.1.1;
  option ip-forwarding off;
  option broadcast-address 192.168.1.255;
  option subnet-mask 255.255.255.0;
}
```

만일 여러분이 리눅스 박스를 캐싱 도메인 네임 서버(caching domain name server)로 설정하고자 한다면, 다음과 같은 옵션을 추가하라:

```
option domain-name-servers 192.168.1.1;
```

If you know your outside DNS addresses and you are *not* going to use the Linux box for DNS, insert the following option, where x.x.x.x and y.y.y.y are IP numbers of the DNS servers: 만일 여러분이 외부의 DNS 주소를 알고 있고 DNS를 위해 리눅스 박스를 사용하지 *않을* 것이라면, 다음과 같은 옵션을 추가하라. 여기서, x.x.x.x와 y.y.y.y는 DNS 서버의 IP 번호이다.

```
option domain-name-servers x.x.x.x, y.y.y.y;
```

만일 여러분이 윈도우즈(Windows) 컴퓨터를 위해 리눅스 박스에 삼바(Samba) 파일 공유를 실행할 것이라면, 리눅스 박스를 기본적인 WINS와 브라우징(browsing) 서버로 사용하도록 다음과 같은 옵션을 추가하라:

```
option netbios-name-servers 192.168.1.1;
option netbios-dd-server 192.168.1.1;
option netbios-node-type 8;
option netbios-scope "";
```

삼바와 WINS를 구성하는 것은 이 문서의 범위를 벗어난다. 만일 몇 가지 지침이 필요하다면 SMBHOWTO로 시작하고, 거기서부터 계속하라.

아직도 두 서너 단계가 더 남아 있다. 다음에, /etc/rc.d/init.d/dhcpd 파일을 편집하여 다음과 같은 행을 찾는다.

```
/sbin/route add -host 255.255.255.255 dev eth1
```

윈도우즈 DHCP 클라이언트는 DHCP 응답에 있어 특이한 브로드캐스트 주소를 필요로 하고, 이 명령은 리눅스/TCP 스택이 그것을 만들어 내도록 한다. 만일 여러분이 파일에서 그 행을 찾을 수 없으면 추가하라. 만일 여러분이 그런것과 같은 행을 찾는다면 그것이 참조하는 장치가 eth1인지 확인하라.

다음 단계는 기본 장치로서 eth1을 사용하도록 /etc/rc.d/init.d/dhcpd 파일을 변경하는 것이다. 다음 행

```
daemon /usr/sbin/dhcpd
```

을

```
daemon /usr/sbin/dhcpd eth1
```

으로 대체하라. 이제 DHCP를 시동할 준비가 됐다. 우선 명령을 사용하여 DHCP 서버를 시작한다: /etc/rc.d/init.d/dhcpd start

마지막으로, DHCP 서버가 리부트(re-boot) 시에 시작할 것인지 확인하여야 한다. DHCP 서버의 어떤 RPM 패키지는 매번 서버가 시작하는 것을 보증하는 지시를 포함하지 않는다. 그래서, chkconfig dhcpd on 명령을 행함으로써 서버가 시작하도록 보장할 것이다.

이 명령으로 인해 레드햇은 /etc/rc.d 아래의 다양한 실행수준(runlevel)의 디렉토리에 DHCP 시동 스크립트를 추가한다. 실행수준 3과 5(multiuser console과 multiuser X)에서는 DHCP 서버가 시작된다. 실행수준 0,1,6(shutdown, single user, reboot)에서는 DHCP 서버는 멈춘다.

3.2.3 클라이언트 컴퓨터

만일 여러분이 DHCP를 설정하였다면, 클라이언트 컴퓨터를 구성하는 것은 매우 쉽다: 단지 DHCP 구성을 가능케만 한다. 윈도우즈 컴퓨터에 대해 이것은 "제어판", 그 다음에 "네트워크" 옵션을 여는 것을 포함한다. "TCP/IP" 프로토콜을 찾고, "구성"하도록 선택한다. "자동으로 IP 주소를 받기"에 해당하는 박스를 체크하고, 변경을 적용하고, 리부트 한다.

리부트 하기 전에 다음과 같은 명령을 입력해 보고 싶을지 모르겠다: tail -f /var/log/messages 이것은 리눅스 시스템 로그를 계속하여 지켜볼 것이다. 만일 모든 게 정상적이면, 윈도우즈 컴퓨터를 리부트 할 때,

IP 주소를 요청한 것을 보게 될 것이고, DHCP 서버가 응답하는 것을 보게 될 것이다. Control-C는 `tail -f` 명령을 종료한다.

만일 여러분이 DHCP를 설정하지 않았다면, 구성은 역시 상당히 쉽다. 다시 "제어판"으로부터 "네트워크" 옵션을 열고, TCP/IP 프로토콜을 구성하기 위해 선택한다. 여러분은 클라이언트 컴퓨터한테 192.168.1.0(네트워크 주소), 192.168.1.255(브로드캐스트 주소) 또는 192.168.1.1(여러분의 리눅스 서버)를 제외한 192.168.1.0 네트워크 내의 어떤 주소든지 할당할 수 있다. 결코 두 대의 컴퓨터한테 동일한 주소를 부여해서는 안 된다. "게이트웨이" 주소를 192.168.1.1에 설정하라. 그래서, 외부로 나가는 트래픽(traffic)이 리눅스 게이트웨이를 통해 보내진다.

The `IPMasqueradingHOWTO` has very detailed information on client configuration in the `ConfigurationSection`. `IPMasqueradingHOWTO`는 `ConfigurationSection`에 클라이언트 구성에 관해 매우 상세한 정보를 가진다.

일반적으로, 클라이언트 컴퓨터를 구성하기 위해서 DHCP 구성을 가능케 하거나 192.168.1.1의 게이트웨이를 가지는 192.168.1.X 네트워크에서 주소를 수동으로 할당한다. 만일 여러분이 캐슁 DNS 서버(아래를 보라)를 구동하거나 DNS를 네트워크 제공자에 의해 할당된 주소로 돌린다면 DNS 서버는 둘 다 192.168.1.1이 되도록 한다.

3.2.4 DNS 서버

리눅스 박스를 캐싱 DNS 서버로 설정하는 것은 넷서핑 속도를 (조금) 향상시킬 것이다. 대개 사용되는 DNS 주소는 여러분의 네트워크 내부에서 캐쉬 되어지고 외부로부터 가져올 필요가 없기 때문이다.

만일 여러분이 완전한 기능을 갖춘 DNS에 관심이 있다면 복잡한 것들을 수많이 배워야 한다. `DNSHOWTO`가 이용 가능하고, `DNSandBIND`책은 좋은 (그리고 매우 이해력이 좋은) 참고 문서이다.

여러분의 클라이언트 장비들을 캐슁 서버로 이용하기 위하여 리눅스 게이트웨이를 그것들의 주 DNS 서버로 사용하도록 반드시 구성되어져야 한다. 섹션 3.2.2에서 보여진 DHCP 지시자는 이를 성취하는 한 가지 방법이다. 만일 여러분이 클라이언트 컴퓨터를 직접 손수 구성하고 있다면, 장비의 IP 주소를 지정하기 위해 사용한 동일한 콘트롤 탭으로 DNS 구성을 변경 가능하다.

DNS 서버를 설치하기 위하여 먼저 `bind` RPM을 설치한다. 그 다음에 `caching-nameserver` RPM을 설치한다. 이 시점에서, 여러분은 거의 준비됐다.

설치됨에 따라, 캐슁 서버는 제대로 작동할 것이다. 그러나 만일 여러분이 인터넷 제공자의 DNS 서버의 IP 주소를 안다면, `/etc/named.conf` 파일을 편집하여 `directory` 행 이후에 다음의 행을 추가함으로써 성능을 조금이라도 향상시킬 수 있다 (여기서, x.x.x.x와 y.y.y.y는 주 DNS 서버와 보조 DNS 서버이다):

```
forwarders { x.x.x.x; y.y.y.y; };
```

이 변경은 여러분의 DNS 서버가 주어진 주소의 검색에 있어 인터넷을 샅샅이 뒤지기 전에 우선적으로 ISP의 DNS 서버를 조회하도록 한다. ISP 서버는 종종 DNS 정보를 충분히 캐쉬 하고, 여러분의 서버가 할 수 있는 것보다 훨씬 빠른 응답을 제공한다.

`named` 데몬은 지난 12 개월을 지나는 동안 몇 가지 보안 문제를 안고 있었다. 그래서, 여러분은 가장 최신 버전을 구동하고 있어야 하고, 보안을 강화하기 위해 기본 설정에 몇 가지 변경을 가하는 것이 매우 중요하다.

1. `bind`의 버전을 점검하고 적어도 8.2.2가 되는지 확인한다. `RedHatUpdates` 또는 `MandrakeUpdates` 사이트로 가서 가장 최신 버전을 점검하라.
2. `/etc/named.conf` 파일에 `forwarders` 행 이후에 `allow-query { 192.168.1/24; 127.0.0.1/32; };` 행을 추가함으로써 로컬 네트워크만에 한해 여러분의 네임 서버에 접근을 제한하라.

3. 여러분의 네임서버를 root로 구동하는 것을 피하라. 만일 여러분의 서버가 root로 구동 중이라면, 서버의 익스플로잇(exploit)이 이기적인 이용자한테 root 권한을 허용하게 될 것이다. 만일 여러분이 서버를, nobody와 같이, 무능한 사용자로 구동한다면, 네임 서버 익스플로잇의 위험을 줄일 수 있다. 여러분의 네임 서버를 nobody로 구동하려면, /etc/rc.d/init.d/named 파일을 편집하여 daemon named 행을 daemon named -u nobody -g nobody로 고친다.

여러분의 DNS 서버가 부트 시에 시작할 것인지 확인한다: chkconfig named on. 다시, 이는 서버는 부트 시에 보통의 실행수준(3과 5)에서 시작될 것이라는 것을 보증한다.

자, 이제 여러분은 DNS 서버를 시작할 수 있다: /etc/rc.d/init.d/named start

3.2.5 내부 네트워크를 시험하기

외부 네트워크를 구성할 때까지는 (인터넷 상에서 다른 DNS 서버와 연결되어야 하기 때문에) DNS 서비스는 작동하지 않을 것이다. 그러나, ping 프로그램을 사용하여 기본적인 내부 연결 상태를 검사할 수 있다.

여러분의 클라이언트 컴퓨터 중 한 대에서 터미널 (MSDOS) 창을 열고, ping 192.168.1.1를 입력한다. 이는 패킷을 규칙적인 간격으로 여러분의 리눅스 컴퓨터로 송출하고 여러분의 리눅스 컴퓨터는 패킷을 되돌려 주게 될 것이다. 만일 제대로 작동 중이면, 한 벌의 패킷 회신을 수 차례 보아야 한다.

3.3 외부 네트워크를 구성하기

이제 기꺼이 외부 네트워크를 구성할 수 있다. 인터넷 제공자가 리눅스를 얼마나 제대로 지원하는지에 따라, 때때로 이것은 어려울 것이다. ADSL 이슈를 다소 자세히 취급하는 ADSLmini-HOWTO이 있다. 만일 내가 Cable Modem HOWTO를 찾을 수 있으면, 또한 그것을 링크로 만들겠다.

대부분의 외부 연결에 관한 주요 문제는 *IP 주소를 얻는 것*이다. 어떤 인터넷 제공자는 케이블 또는 ADSL 가입자한테 정적인 IP 주소를 나누어 준다. 그 경우에 구성은 쉽다. 하지만, 대부분의 제공자들은 DHCP를 통한 동적인 구성으로 바꾸었다. 이것은 여러분의 리눅스 컴퓨터가 여러분의 eth1 인터페이스 상에서 DHCP *서버*, 여러분의 eth0 인터페이스 상에서 DHCP *클라이언트*가 되기 쉬울 것이라는 것을 의미한다.

Additionally, many providers have taken to providing their services in specialized non-standard ways which assume their customers will be using Windows. Some of those cases will be discussed at the end of section 3.3.2. 게다가, 많은 제공자들은 그들의 고객들이 윈도우즈를 사용 중일 것이라고 가정하는 특수화된 표준이 아닌 방식으로 서비스를 제공하는 것에 따르고 있다. 그런 경우들의 몇 가지는 섹션 3.3.2의 끝 부분에서 논의될 것이다.

3.3.1 정적인 IP를 가진 경우

만일 여러분의 인터넷 제공자가 여러분한테 정적인 IP 주소를 할당한다면 여러분은 유리한 처지에 있다. 먼저, 새로운 인터페이스 구성 파일을 생성하고, /etc/sysconfig/network-scripts/ifcfg-eth0, 다음을 그 안에 넣는다.

```
DEVICE=eth0
IPADDR=x.x.x.x
NETMASK=y.y.y.y
ONBOOT=yes
```

여러분의 인터넷 제공자에 의해 주어진 값으로 x.x.x.x와 y.y.y.y를 채우기만 한다. 이제 /etc/resolv.conf 파일을 편집하여 다음 정보를 입력한다.

```
search provider_domain_here
nameserver n.n.n.n
nameserver m.m.m.m
```

provider_domain은 인터넷 제공자에 의해 공급되어야 한다. 또한, 주 DNS 서버와 보조 DNS 서버를 n.n.n.n과 m.m.m.m 행에 입력한다. 만일 여러분이 리눅스 박스를 DNS 서버로 설정하였다면 다른 네임서버 기입사항 이전에 하나의 행을 추가한다: `nameserver 127.0.0.1`. 이것은 외부 서버한테 DNS 정보를 요청하기 전에 여러분의 리눅스 서버가 캐싱 서버를 사용하도록 한다.

3.3.2 DHCP를 사용하는 경우

만일 인터넷 제공자가 DHCP 구성을 사용한다면, 새로운 인터페이스 구성 파일을 생성하고, `/etc/sysconfig/network-scripts/ifcfg-eth0`, 다음을 그 안에 넣어 둘 필요가 있다:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

이제 dhcpcd 클라이언트 데몬이 여러분의 시스템에 설치되었는지 확인한다. 리눅스 CD로 가서 `dhcpcd` RPM 패키지를 설치한다.

여러분의 새로운 네트워크 구성을 시험할 때이다. `/etc/rc.d/init.d/network restart` 명령을 사용하기만 하라. 이제 ping을 사용하여 여러분의 외부 연결을 시험하라. `www.yahoo.com`과 같은 인터넷 상의 컴퓨터를 핑(ping) 하고 어떤 것이 되돌아 오는지 보아라.

3.3.3 기발함과 예외들

여러분의 상황은 위에서 기술된 매우 간단한 상황과는 다를지 모른다. 여기에 가지각색의 차이에 관한 몇 가지 간결한 견해와 좀 더 믿을만한 자원이 있다. 링크와 이 섹션을 추가하도록 자극을 제공한 데 대해 John Mellor한테 감사한다.

이더넷 상에서 PPP (PPPoE) 다수의 ADSL 제공자들(예를 들어, Bell Atlantic)은 새로운 고객은 "PPP over Ethernet" 프로토콜 (PPPoE)을 사용하여 서비스에 연결하여야 한다고 강조하고 있다. 이 목적을 위해 그들은 윈도우즈 클라이언트 프로그램을 제공하지만, 리눅스 사용자들한테는 그다지 유용하지 못하다. 다행스럽게도, PPoE는 간단한 프로토콜이고, 리눅스 하에서도 지원하기 위해 다수의 노력이 진행 중에 있다.

- `RoaringPenguinPPPoEClient`는 독자인 Kerr First에 의해 높이 평가되어 추천되어졌다.
- `PPPoEonLinuxforBellSympatico`
- PPPoE on Linux for Sympatico (`GeneralInfo`) (`LinuxInfo`)

바보같은 DHCP 트릭 네트워크 제공자가 행하는, 특히 좋아하는 트릭들 중 하나는 여러분의 서비스를 고유한 호스트 명칭, 또는 심지어 고유한 네트워크 카드에 연관시키는 것이다. 이것은 허브를 사용하여 여러 대의 컴퓨터가 여러분의 이더넷 포트 내로 꽂는 것으로부터 여러분을 그럴 듯하게 보호한다. (물론, 여러분은 리눅스와 매스커레이딩을 사용함으로써 더 나은 보안을 가지는 동일한 효과를 얻게 될 것이고 케이블 회사는 어떤 방법으로도 알지 못한다!)

만일 제공자가 여러분한테 호스트 명칭을 부여하고 그들의 서비스를 사용하기 위해 여러분의 윈도우즈 박스를 그 명칭으로 설정할 것을 강조한다면, 여러분은 DHCP 서버로부터 주소를 요청할 때 여러분의 리눅스 박스가 그 호스트 명칭으로 전송하는지 확인하여야 할 것이다.

인터페이스 구성 파일에서 BOOTPROTO를 dhcp에 지정할 때 레드햇 DHCP 클라이언트가 호출된다. 그러나 호스트 명칭에 참조하지 않고 호출된다. 호스트 명칭을 사용하여 프로그램을 호출하기 위해서, 레드햇 6.1에서, `/etc/sysconfig/network` 파일을 편집하여 다음 행을 변경하라:

```
HOSTNAME=
```

이를 판독하려면:

```
HOSTNAME=your_isp_assigned_name
```

이것은 레드햇 변종의 어떤 것에서는 작동하지 않을 지도 모른다. 만일 작동하지 않으면, `/sbin/ifup` 스크립트를 점검하고 dhcpcd와 pump를 호출이 -h $HOSTNAME 스위치를 포함하는지 보라. 만일 그렇지 않으면, 그것들을 추가하라. 그래서 호출은 `/sbin/dhcpcd -i $DEVICE -h $HOSTNAME`와 `/sbin/pump -i $DEVICE -h $HOSTNAME`과 같다.

Road Runner Road Runner 케이블 서비스는 서버가 사용될 수 있기 전에 반드시 실행되어야 하는 특수한 로그인 처리를 가진다. 다행스럽게도, 상세한 `LinuxRoadRunnerHOWTO`가 이용 가능하다.

3.3.4 네트워크 엔트리를 보기

이제 여러분의 작품에 감탄할지 모르겠다. 모든 구성된 장치를 보기 위해 `ifconfig`을 입력하라. 나의 게이트웨이 컴퓨터에서 이와 같이 보인다:

```
eth0  Link encap:Ethernet  HWaddr 00:60:67:4A:02:0A
      inet addr:24.65.182.43  Bcast:24.65.182.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
      RX packets:487167 errors:0 dropped:0 overruns:0 frame:0
      TX packets:467064 errors:0 dropped:0 overruns:0 carrier:0
      collisions:89 txqueuelen:100
      Interrupt:10 Base address:0xe400
eth1  Link encap:Ethernet  HWaddr 00:80:C8:D3:30:2C
      inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
      RX packets:284112 errors:0 dropped:0 overruns:0 frame:1
      TX packets:311533 errors:0 dropped:0 overruns:0 carrier:0
      collisions:37938 txqueuelen:100
      Interrupt:5 Base address:0xe800
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:3924  Metric:1
      RX packets:12598 errors:0 dropped:0 overruns:0 frame:0
      TX packets:12598 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
```

`eth0` 인터페이스는 색다른 외부의 IP 주소를 가지고, `eth1` 주소는 개인용 내부 주소를 가진다.

`route` 명령을 입력함으로써 네트워크 라우터를 볼 수 있다. 나의 게이트웨이 컴퓨터에서 이와 같이 보인다:

```
  Kernel IP routing table
```

```
Destination     Gateway      Genmask         Flags Metric Ref Use Iface
255.255.255.255 *            255.255.255.255 UH    0      0     0 eth1
192.168.1.0     *            255.255.255.0   U     0      0     0 eth1
24.65.182.0     *            255.255.255.0   U     0      0     0 eth0
127.0.0.0       *            255.0.0.0       U     0      0     0 lo
default         24.65.182.1  0.0.0.0         UG    0      0     0 eth0
```

여기서 외부 네트워크가 설정된 것을, 내부 네트워크가 설정된 것을, 로컬 장치가 설정된 것을, 특정한 255.255.255.255 브로드캐스트 주소가 설정된 것을, 기본 라우터가 인터넷 제공자 게이트웨이를 가리키도록 설정된 것을 볼 수 있다. 완벽하다!

이제 여러분은 외부 네트워크와 내부 네트워크를 가진다. 남은 전부는 둘 사이를 개통하는 것이다. 그렇지만, 먼저 어떤 괴물도 외부로부터 들어올 수 없어야 한다는 것을 확인한다.

3.4 보안

ADSL 또는 케이블을 경유해 인터넷에 영구적으로 연결되는 것의 결점 중의 하나는 여러분의 컴퓨터가 일주일 7 일간, 하루 24 시간 내내 잠재적인 보안 위협에 노출되어 있다는 것이다. 리눅스를 게이트웨이로 사용하는 것은 모든 다른 컴퓨터를 감추기 때문에 위험을 감소시킨다. 그 밖의 인터넷에 관한 한, 단지 여러분의 리눅스 박스만이 연결을 위해 사용 가능하다. 그래서, 이 시점에서, 여러분의 박스를 보다 더 안전하게 만들기 위한 기본적인 팁을 두 서너 개 알려 줄 것이다.

먼저, 모든 나쁜 놈들을 차단할 필요가 있다. 이것을 하기 위해 `/etc/hosts.deny` 파일을 편집하여 이와 똑같은지 확인하라:

```
#
# hosts.deny  This file describes the names of the hosts which are
#             *not* allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
#            The portmap line is redundant, but it is left to remind you that
#        the new secure portmap uses hosts.deny and hosts.allow. In particular
#             you should know that NFS uses portmap!
ALL: ALL
```

이것은 – 들어오는 연결의 95%를 제어하는 – "TCP wrappers"한테 모든 호스트로부터의 모든 연결을 거부하도록 알린다. 그것은 매우 훌륭한 규칙이다! 그러나 그것은 여러분이 내부의 가정용 네트워크로부터 리눅스 박스에 연결되는 것도 역시 막게 될 것이다. 이는 곤혹스럽다. 그래서, 한 가지 예외를 만들 것이다. `/etc/hosts.allow` 파일을 편집하여 이와 똑같은지 확인하라:

```
#
# hosts.allow  This file describes the names of the hosts which are
#              allowed to use the local INET services, as decided
#              by the '/usr/sbin/tcpd' server.
#
ALL: 127.0.0.1
ALL: 192.168.1.
```

This tells the "TCP wrappers" that they can allow connections to all services from the local device (127.0.0.1) and from your home network (192.168.1.). 이것은 "TCP wrappers"한테 로컬 장치(127.0.0.1)로부터 그리고 여러분의 가정용 네트워크(192.168.1.)로부터 모든 서비스에 연결을 허용할 수 있다는 것을 알린다.

강력한 맹꽁이 자물쇠를 사용하여 여러분은 이제 외부의 괴물을 막았다. 만일 여러분이 가로 빗장과 경보 시스템을 갖추기를 원한다면, 훨씬 더 많은 지식이 있어야 할 것이다. 만일 여러분이 안전한 리눅스 박스에 대해 더 많이 배우기를 원하면 SecurityHOWTO는 시작하기에 좋은 곳이다.

제 4 절 매스커레이딩을 구성하기

예비 단계는 지났다. 여기가 마술이 시작되는 곳이다. IP 매스커레이딩은 리눅스가 제공하는 참으로 마술과 같은 서비스의 하나이다. 동일한 것을 하는 윈도우즈를 위한 상품이 있지만, 도저히 효율적이지 않다: 오래된 386은 명확하게 IP 매스커레이딩 서비스를 전체적으로 중간 정도의 크기를 갖는 사무실에 제공한다. 그러나, 애드온(add on) 매스커레이딩 패키지는 말할 것도 없고, 윈도우즈 95를 실행조차 하지 못한다. (부록 삼아, 몇 가지 최근의 평가기사에서, 나는 윈도우즈 2000이 애드온 소프트웨어가 없이도 "연결 공유"를 지원할 것이라고 읽었다. 연결 공유 소프트웨어를 판매한 회사들이 마이크로소프트 사에 의해 "에워 싸이고 확장되는" 것처럼 보인다. 그렇지만, 나는 386 상에서 윈도우즈 2000 솔루션을 시도할 것을 추천하지 않겠다.)

리눅스는 대단히 기능이 풍부한 방화벽 능력을 가진다. 가장 간단하고 가장 조잡한 가능성 있는 방법으로 그것을 사용하게 될 것이다. 만일 여러분이 숙련자와 같은 방화벽을 구축하는 방법을 배우고 싶다면 이론의 이해를 위해 FirewallingHOWTO를, 리눅스 2.2.X(그리고 확장에 의해 레드햇 6.X)와 함께 선적되는 새로운 ipchains 방화벽 구축 도구에 관한 명령을 위해 IPChainsHOWTO를 둘 다 모두 읽어야 한다. 또한 매스커레이딩에 관해 더 자세한 내용을 가지는 매우 훌륭한 IPMasqueradingHOWTO를 이용 가능하다.

일단 여러분의 내부 및 외부 네트워킹이 사용 가능하다면, 간단한 매스커레이딩을 구성하는 것은 매우 매우 쉽다. /etc/rc.d/rc.local 파일을 편집하여 맨 아래 쪽에 다음 행들을 추가하라.

```
# 1) Flush the rule tables.
/sbin/ipchains -F input
/sbin/ipchains -F forward
/sbin/ipchains -F output
# 2) Set the MASQ timings and allow packets in for DHCP configuration.
/sbin/ipchains -M -S 7200 10 60
/sbin/ipchains -A input -j ACCEPT -i eth0 -s 0/0 68 -d 0/0 67 -p udp
# 3) Deny all forwarding packets except those from local network.
#    Masquerage those.
/sbin/ipchains -P forward DENY
/sbin/ipchains -A forward -s 192.168.1.0/24 -j MASQ
# 4) Load forwarding modules for special services.
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
```

마지막 두 행은 FTP와 RealAudio를 허용하는 커널 모듈이 내부 네트워크 상의 컴퓨터를 위해 작동하도록 추가한다. 만일 여러분이 필요하다면, 첨가 가능한 특정한 서비스를 위한 다른 모듈들이 있다:

- CUSeeMe (/sbin/modprobe ip_masq_cuseeme)
- Internet Relay Chat (/sbin/modprobe ip_masq_irc)
- Quake (/sbin/modprobe ip_masq_quake)

- VDOLive (/sbin/modprobe ip_masq_vdolive)

이제 매스커레이딩을 시도한다. /etc/rc.d/rc.local 명령을 사용하여 rc.local 스크립트를 실행하라. 여러분의 다른 컴퓨터들 중의 한 대에 앉아서 웹 서핑을 시도하라. 행운이 따른다면, 모든 것이 이제 나무랄 데가 없다.

제 5 절 Problems 문제점

이와 같은 간단한 문서를 사용하여 잘못 될 수 있는 것들이 무수히 많이 있다. 특수한 경우가 많기 때문이다. 가능성 있는 문제점의 대다수가 내부 및 외부 네트워크 장치의 구성에 원인을 두고 있다. 내가 시도하고 문제점을 가진 사람들한테 응답할 것이다. 무엇이 잘못 되었는지 알아 내고, 특수한 경우의 문제점을 가진 사람들이 도움을 자세히 찾아낼 수 있도록 하기 위하여 이 아래에 링크를 추가할 것이다. pramsey@refractions.net로 나와 연락하는 것을 편하게 여겨라.

5.1 ICQ가 작동하지 않는다

ICQ의 어떤 부분은 매스커레이딩 상에서 제대로 작동한다. 다른 부분은 전혀 작동하지 않는다. 그렇지만, 매스커레이딩 상에서 ICQ를 실행하는 데 부족한 어떤 (그러나 전체가 아닌) 것을 맡아 주는 개발 중인 betaqualityICQmodule이 있다. 소스 코드 배포 파일에 포함된 README 파일은 모듈을 컴파일 하는 방법을 기술한다. 일단 여러분이 컴파일 하고 설치하였다면, /sbin/modprobe ip_masq_icq를 호출하라.

5.2 레드햇 6.X가 아닌 칼데라 2.X를 가지고 있다.

알아 두어야 하는 몇 가지 중요한 변경 사항이 있다:

1. 인터페이스를 위한 /etc/sysconfig/network-scripts/ifcfg-eth0 & eth1에서 GATEWAY=xxx.xxx.xxx.xxx 문장 (로컬 인터페이스는 리모트 인터페이스 IP 주소를 사용하고 리모트 인터페이스는 서비스 제공자의 게이트웨이 IP를 사용한다)
2. /etc/sysconfig/daemons/dhcpd 스크립트가 ROUTE_DEVICE를 eth0이 아닌 eth1로 나열하는지 확인한다.
3. /etc/dhcpd.conf는 인터페이스 둘 모두를 위한 서브넷 문장을 필요로 한다. (나는 내가 두 번째 문장을 만든 바와 같은 이유를 완전히 확신하지 못한다: 어떤 다른 옵션조차 없는 subnet 216.102.154.201 netmask 255.255.255.255 { } DHCP 서버는 eth0와 eth1 뿐만 아니라 폴백(fallback) 상에서 수용하고(listen) 전송한다.)
4. /etc/rc.d/init.d/dhcpd 스크립트에 add host route 255.255.255.255를 추가하지 말아라. 칼데라는 이미 문제점을 고쳤다. 스크립트 내에서 eth0에 참조하는 전부를 eth1로 변경하고 확인하라.

5.3 내부 장비 중의 한 대가 웹 서버가 되기를 바란다

식은 죽 먹기와 같다! 그렇지만, 이렇게 쉬운 안내로 작동하기 위해서는 *여러분은 정적인 IP 주소를 가져야 할 필요가 있다*. 만일 여러분이 동적인 주소를 가진다면, 주소가 변경될 때 여러분의 IP 주소가 포트 전달 명령으로 갱신되어지는 것을 보장하도록 몇 가지 추가적인 스크립트를 조작할 필요가 있다.

명심하라, 외부 포트를 내부 장비로 전달하는 것은 여러분의 "내부" 장비를 이전보다 덜 "내부"적으로 만든다는 것을. 그러나 매우 명백하게 받아 들여질 수 있고, 거의 없거나 아예 없는 성능 degredation을 마치게 된다. 리눅스 커널에 있는 IP 매스커레이딩 코드의 부작용의 하나는 네트워크 레이어(network layer)에

당도할 때 패킷을 사용하여 몇 가지 상당히 재미있는 일을 하는 능력이다. ipmasqadm 유틸리티는 그런 능력을 이용하기 위해 제작되었다.

몇 가지 이유 때문에 ipmasqadm이 모든 레드햇과 맨드레이크 변종과 함께 선적되지 않는다. 그래서 유지관리자의 로부터 가져 와야 할 것이다. – 이용 가능한 RPM 뿐만 아니라 소스 코드도 있다.

일단 여러분이 RPM을 가지고, 설치하라. 그 다음에 /etc/rc.d/rc.local 파일에 다음의 행들을 추가하라:

```
/usr/sbin/ipmasqadm portfw -f
/usr/sbin/ipmasqadm portfw -a -P tcp -L x.x.x.x 80 -R 192.168.1.x 80
```

첫 번째 명령은 포트 전달 규칙을 강제로 내보내고, 두 번째 명령은 외부 인터페이스 상의 포트 80으로부터 내부 장비 상의 포트 80으로 전달을 추가한다. 외부의 정적인 IP 주소는 x.x.x.x 자리에 들어 가고, 내부 장비 IP 주소는 192.168.1.x 자리에 들어간다.

이제 포트 80을 위한 외부의 요청은 명백하게 내부 장비의 포트 80으로 보내어질 것이다. 여러분은 내부 장비 중 한 대로부터 여러분의 게이트웨이의 포트 80에 텔넷을 하거나 연결하는 것으로는 이것을 시험하지 못한다는 것을 주의하라: 포트 전달자는 *외부* 인터페이스에서 들어오는 요청을 그저 이행할 뿐이다.