Domain Mini-HOWTO v0.6

크리스토퍼 뉴펠드*neufeld@physics.utoronto.ca* <mailto:neufeld@physics.utoronto.ca> 2000년 1월 9일

번역: 전혜진*hjchj@nownrui.net* <mailto:hjchj@nownrui.net>

이 문서는 당신이 자신의 도메인으로 네트웍을 구성하기를 원할 때 대개 해야만 하는 일들에 대한 개요라 할 수 있다. 그것은 네트웍 서비스와 보안 등등에 관한 것을 포함한다.

목 차

제 1 절 들어가기 전에.....

1.1 이 문서에 의한 책임을 지지 않음.

이것은 거의 초고에 가까운 문서라 할 수 있다. 나는 더 자세한 많은 것들을 용케 배제한 경향이 있다는 것과, 중요한 것을 놓쳤을 지 모른다는 것을 미리 밝혀 두겠다. 나는 이 문서가 더욱 널리 이용할 수 있는 것으로 하기 위해 사람들이 추가할 내용이나 더 많은, 혹은 짧은 것이라도 세부적인 사항을 보내 주길 바란다.

1.2 저작권

제 2 절 시작하기

이것은 고정 IP나 도메인 이름이 붙어 있는 서버에 지속적으로 연결되어 있는 리눅스 머신이나 혹은 리눅스와 윈도 환경이 섞여 있는 환경에서 당신의 도메인으로 네트웍 설정을 하는 것의 안내문이다. 이 글은 유동 IP를 사용하거나 때때로만 네트웍에 연결되는 컴퓨터에는 그리 도움이 되지 않을 것이다. 그러나 그런 사용자들을 위해 "유동 IP 사용하기" 섹션에 몇 가지 기본적인 도움말을 수록해 놓았다.

고정 IP주소로 된 컴퓨터들을 몇 대 더 연결하는 경우 실 도메인의 셋업 은 사람에게도 그 구성되는 컴퓨터 자체에도 인터넷에 연결하는 것이 더욱 쉬운 일이 된다. 처음 시작할 때 계획적으로 일을 하는 것은 향후에 일어날 문제들을 줄이는 가장 좋은 방법이라 할 수 있을 것이다.

이 문서의 대부분이 기술하는 것은 새로 구성되는 네트웍의 기본적인 보안에 관한 것이다. 이러한 것들은 외부의 공격 혹은 뜻하지 않은 내부의 공격에 대응하기 위한 것이다. 그것은 완벽한 보안의 예방책을 주장하는 것은 아니 지만, 일반적으로 공격자로부터의 피해를 줄이는 데는 충분하다.

이 문서는 1차적인 타겟으로 구성을 운직이기 쉽고 빠른 연결 속도를 보이는 다이얼업 연결이 가능한 소규모의 네트웍상의 컴퓨터 혹은 외부로 데이터를 내보내는 것을 향상시키려 하거나 혹은 WWW, FTP 사이트를 운영하려 하는 컴퓨터로 잡고 있다. 또한 이 문서는 새로이 자신의 도메인으로 네트웍을 구축하려 하면서 속도와 성능의 향상을 개선하려 하는 경우에도 유용할 수 있을 것이다.

이 문서를 통해 나는 새로운 등록된 도메인으로 구성하는 법을 논할 것이다. example.com 과 같은 것 말이다. 이름의 예로 든 example.com 에 대해 간단하게 말하자면, 등록된 IP주소를 문자열로 부르는 것으로, 실재하는 도메인이 아니라는 것만 밝혀 두겠다.

이 문서 안에 있는 대부분의 정보는 다른 곳에서도 얻을 수 있는 것들이다. 나는 새로운 도메인의 설정에 있어 기본적으로 적절한 것들만을 뽑아내려 노력하였다. 세부적이고 특수한 경우에 관한 것은 부족한 점이 많으므로 아마 당신은 한개 이상의 다른 세부적인 문서를 참고해야 할 지도 모르겠다.

이 문서는 여러 OS 를 함께 사용하는 경우에 대해서도 다루고 있다. 보다 엄밀히 말하자면 나는 마이크로소프르의 윈도의 어느 버전을 사용하는 데스크탑 환경에서 서버와 게이트웨이로서 리눅스를 사용할 경우를 다룰 것이다.

제 3 절 네트웍 구성을 계획하기

수많은 다른 모습의 네트웍 구성 방식이 있겠지만,. 많은 구성방법이 택하고 있는 것은 데스크탑 머신들이 놓여 있고 서버들이 매스커레이드 서브넷에 자리하고 있으며 일반적인 접근이 가능한 컴퓨터들이 유효한 대외적인 IP를 갖고 있는 형태일 것이다.유효한 대외적인 IP를 갖고 있는 컴퓨터들은 이 문서 에서는 “노출된 호스트”라는 명칭을 사용하여 나타낼 것이다. 다음은 일반적인 예가 될 구성 방식이다.

```
+--------------+
|              |               +---------------+
| ISP-supplied |---------------| FTP server    |
| router       |        |      +---------------+
|              |        |
+--------------+        |      +---------------+
                        |------| WWW server #1 |
                        |      +---------------+
                        |
                        |      +---------------+
                        |------| WWW server #2 |
                        |      +---------------+
                        |
                        ~
                        ~
                        |
                        |      +---------------+
                        |------| Private       |
                               | Network       |
                               | Gateway       |
                               +---------------+
                                      |
                                      |
                                      |
                                      |
     +------------+                   |      +-------------------+
     | Desktop #1 |-------------------|------| Private server #1 |
```

```
+------------+                   |      +-------------------+
                                 |
        .     -------------------|--------        .
        .                        |                .
        .     -------------------|--------        .
                                 |
+------------+                   |      +-------------------+
| Desktop #N |-------------------|------| Private server #N |
+------------+                          +-------------------+
```

이 예에서 라우터는 ISP (인터넷 서비스 공급자), FTP 서버, WWW 서버, 그리고 "공개되지 않은 네트웍 게이트웨이" 라고 불리는 컴퓨터에게 외부로 나가는 IP 번호를 공급받는다. 그리고 데스크탑이나 사설 서버의 IP는 RFC 1918 *www.ietf.org/rfc/rfc1918.txt* <http://www.ietf.org/rfc/rfc1918.txt>에 의해 공급된다. 당신이 사설 네트웍(모든 컴퓨터들이 사설 게이트웨이 아래에 구축되는)에 사용하기 위해 선택한 IP 번호는 당신이 관리하는 호스트 아래에 있는 다른 어떤 컴퓨터도 사용하지 않는 유일한 것이 되어야 한다. 그러나 각각의 독립되어 있는 서브넷이나 가상적인 네트웍일 경우에는 그리 되더라도 충돌을 일으키지는 않을 것이다. 네트웍을 합병할 때도 그런 식으로 재배열 하면 될 것이다. RFC의 개요는 당신이 192.168.0.* 에서 시작하여 192.168.255.* 로 끝나는 어떤 C 클래스의 네트웍이나 혹은 172.16.*.* 에서 172.31.*.*로 가는 B 클래스의 네트웍, 아니면 10.*.*.*의 A 클래스의 주소를 할당하는 데 있다. 이 문서의 나머지 부분은 당신이 만들기를 선택한 C 클래스의 사설 네트웍을 설명할 것이다. 그리고 인터넷 공급자가 공급한 IP 번호 중 하나인 IP 번호가 10.1.1.9인 네트웍 게이트웨이에 관한 것도 포함된다. (좀 전에 예로 든 것은 유효한 IP가 아닌 그저 예제로 사용하는 번호임을 미리 밝혀 둔다.) 나는 10.1.1.10의 IP를 가지고 웹서버와 FTP 서버를 겸하는 betty.example.com에 대해서도 다룰 것이다.

당신의 컴퓨터에 필요한 외부 IP 번호에 관해 메모하라. 당신은 각각의 컴퓨터마다 외부의 다른 컴퓨터와 구별되는 하나씩의 IP주소가 필요할 것이다. 이와 같은 셈은 라우터에 의해 주어졌거나 내부에서만 통용되는 IP 번호들을 포함하지는 않는다. 당신은 당신의 인터넷 공급자에게서 각각의 기계에 부여하기 충분할 정도로 많은 갯수의 IP번호들을 얻을 수 있을 것이다. 예를 들면, ISP로부터 8개의 IP를 공급받은 나의 사무실 네트웍에서 나의 컴퓨터 중 3대는 4대의 밖으로 나가는 게이트웨이에만 충분했을 뿐이었던 IP 주소 때문에 사용 불능이어서 게이트웨이 자신에 추가할 수 밖에 없었다.(문맥이 매끄럽지 않군요..... 이런...... T.T)

이 네트웍 구조는 모든 이에게 옳은 것은 아니다. 그러나 그것은 특별한 경우를 제외한 대부분의 설정에서 합당한 출발점이라 할 수 있다. 이러한 설정을 채택할 때의 이점은 다음과 같다.

? 확장이 용이하다. 만일 당신이 곧 당신의 노드를 두 배로 확장할 계획 이라면 당신은 인터넷 공급자로부터 새로운 IP 블록을 얻는다던가 당신의 컴퓨터의 인터페이스를 완전히 재설정할 걱정은 안 해도 될 것이다.

? 지역 네트웍의 관리시. 새로운 워크스테이션을 당신의 네트웍에 당신의 인터넷 공급자와의 커뮤니케이션 없이 추가하기를 원할 때를 생각해 보자. 필요한 일들(ssh나 ftpd는 DNS와의 연결된 피드백이 없이는 불평을 하게 된다.)을 할 때 마다 DNS(도메인 네임 서버)와의 함수적인 연결 관계가 필요한 공개된 노드와는 다른 일이 되겠다. 역전된 DNS 쿼리는 IP 번호에서부터 호스트의 이름을 얻어내게 된다.

주요 보안에 관하여. 사설 네트웍 게이트웨이는 그 네트웍에 연결된 각각의 데스크탑에 어떤 표준적인 것을 인스톨하는 대신에 전체의 사설 네트웍에 관하여 패킷을 필터링하거나 로그인 공격에 대한 보안을 실시할 수 있다. 이것은 내부로 들어오는 패킷에 관한 것만이 아닌 밖으로 나가는 패킷에도 해당되는 것이다. 따라서 설정되지 않은 데스크탑은 인터넷이라 불리우는 외부를 향해 데이터를 내보낼 수도 없는 것이다.

? 이동의 용이성이 또 다른 중요한 문제이다. 당신의 네트웍 안의 IP 주소들은 당신이 원하는 한 언제까지라도 당신 고유의 것들이다. 당신은 사설 네트웍의 설정을 변경하지 않고도 전체 네트웍을 새로운 IP 주소값으로 변경할 수 있다. 따라서 공공연히 위험에 노출된 호스트들은 새로이 설정되어야 할 것이다.

인터넷 연결의 투명성. 당신의 사설 네트웍에 연결된 컴퓨터들은 FTP, telnet, WWW 그리고 약간의 장애를 동반한 채로 리눅스 매스커레이딩 라우터가 떠맡는 다른 서비스들을 이용할 수 있다. 사용자들은 그들

의 컴퓨터가 외부적에서 보이지 않는 IP 번호라는 사실을 깨닫지 못하면서 말이다.

이런 식으로 설정을 할 때의 불편함으로 대두될 가능성이 있는 것들은 다음과 같다.

어떤 서비스는 내부 네트웍에 연결된 컴퓨터에서 즉각 이용할 수 없게 될 지도 모른다. 외부 호스트에 대한 NTP 의 일치는 커널의 매스커레이딩 규칙을 포함하지 않는 서비스를 어느정도 덮어 감추게 한다. 또한 .shost 인증은 외부의 노드로부터의 접근은 어렵게 혹은 불가능하게 하지만, 내부에서는 거의 언제나 가능하게 한다.

네트웍 하드웨어에 관한 경비가 더 많이 든다. 사설 네트웍 게이트웨이 는 2개의 네트웍 카드와, 외부로 노출된 네트웍과 사설 네트웍을 위한 최소 2개의 허브, 스위치를 필요로 하게 된다.

?사설 네트웍 안의 컴퓨터가 그 외부의 컴퓨터들과 한 번에 연결되게 하는 것은 쉬운 일이 결코 아니다. 외부의 컴퓨터들과 연결될 때 맨 처음 일어나는 일은 먼저 네트웍 게이트웨이를 지나며 외부 호스트로의 연결에 대한 로그를 남기는 것이다. 이것은 라우터의 패킷들이 투명성을 유지하며 방화벽을 지나는 것으로 가능해 지지만, 그것은 보안상의 문제에 대한 충고를 배제 한다. 이에 관한 것은 다음 섹션에서 마저 논하겠다.

당신은 당신의 네트웍 구성에 있어 그러한 점들을 숙고하고 당신의 상황에 꼭 적당하게 외부로 보이는 네트웍을 결정해 낼 수 있을 것이다. 이 문서의 나머지에서 나는 당신의 네트웍에서 보여지는 것 이상의 설정에 관해 논할 것이다. 만약 당신이 당신에게 꼭 맞는 네트웍을 구성하기로 결정했다면, 세부적인 다른 부분에 관해 나는 이 문서에세 추가로 서술하도록 할 것이다.

특별한 케이스로, 만약 당신이 외부로 보여지는 서버는 아예 필요가 없다면, ISP 서플라이드 라우터는 당신의 사설 네트웍 게이트웨이에 있는 외부 인터페이스에 (허브보다 나은 대안으로)바로 소속될 수 있을 것이다.

제 4 절 연결하면서

4.1 인터넷 공급자를 선택하기

주위에서 얻을 수 있는 어떤 것이라도 좋을 것이다. 당신에게 서비스를 공급할 수 있는 업체 중에 다음의 서비스들을 제공하는 업체로서 가격이 적당한 곳을 고르도록 하라. DSL 지원이 모든 곳에서 이루어지지는 않고 어떤 지역에서는 무선 연결 지원이 지형이나 건물 혹은 환경의 영향으로 적당하지 않을 수 있다. 당신의 배선이 놓을 곳의 주소에 따라 공급될 것을 각오하라. DSL의 속도는 스위치와의 거리에 반비례한다. 그리고 당신의 컴퓨터와 공급자 사이의 대역폭과, 연결 설정의 종료법, 그리고 매달 당신의 컴퓨터를 인증하는 하드웨어 등등을 따로 세세히 물어보도록 하라. 그리고 당신은 얼마나 많은 IP주소를 당신의 컴퓨터들에 할당할 수 있을지도 알아 두어야 할 것이다. 당신이 공급자에게서 얻는 IP 주소야말로 당신이 당신의 컴퓨터에 접근할 수 있는 주소라는 것을 기억하도록 하라. 당신의 사이트와 또 다른 곳들로 연결되는 속도인, 외부로 연결되는 총 대역폭을 질문하도록 하라. 만일 공급자가 외부로의 부족한 대역폭을 갖고 있다면, 소비자는 병목 현상을 경험하게 될 것이다.

공급자들 중에서 선택의 폭을 좁혀 나갈 때는, 주위에 여러 모로 물어 보고 주위 사람들이 추천하는 업체를 고려하도록 하라. 대역폭이 큰 순서대로 알아보도록 하고, 새로운 도메인과 지역 ISP사이의 집에서의 접속 속도를 텔레커뮤니케이션을 위해 혹은 원격 관리를 위해 확인할 생각이라면 집에서 ISP 로 연결을 하여 당신이 접속하고 싶은 곳까지의 traceroute 를 실행해 보는 것이 필수이다. 이것은 그 경로에 있어 얼마나 많은 허브를 지나가게 되는지, 어느 정도의 지연이 있을지를 말해준다. 100에서 200 밀리세컨드(천분의 1초 단위) 이상의 지연은 시간 경과를 멈추기 어렵게 할 것이다. traceroute는 당신이 당신의 새로운 네트웍 환경에서 당신의 집에서 새로운 도메인까지의 지연 시간을 파악하게 할 수 있다.

4.2 하드웨어 설치 준비

당신이 새로운 도메인의 설정을 위한 공급자와 서비스 방식을 선택했다면 설치의 세부적인 면을 질문하도록 하라. 당신은 서비스를 시작하기 위해 서비스를 제공하는 ISP에 전화로 연락을 하여 서비스를 요청하고 당신의 네트웍이 설치될 빌딩의 제어와 접속을 위한 기술자를 부를 수 있다. 그렇게 하여 빌딩의 엔지니어에게 요구 사항을 알릴 수 있게 되는 것이다.

ISP의 기술자가 오기 전에 네트웍 파라메터와 IP번호, 넷마스크, 브로드캐스트와 게이트웨이 라우터, DNS 서버의 주소, 그리고 당신의 컴퓨터들을 연결하여 넘겨줄 배선에 관해 질문하도록 하라. (예를 들자면, 일자로 쭉 뻗은 :straight- through 이라던가 혹은 교차된:crossover RJ45 배선이라던가 하는 식으로.....)

테스트를 위한 컴퓨터를 준비하고, 네트웍 연결 설치가 될 근처에 두어라. 가능하다면, 서비스 기술자가 도착하기 전에 IP 번호와 넷마스크를 설정하고 적당한 배선을 갖추어 보아 설치와 테스트가 빨리 끝날 수 있도록 하자.

4.3 연결 테스트

테스트할 컴퓨터를 ISP 의 하드웨어에 연결하라. 이 때 ISP 밖의 사이트로 ping을 해야 한다는 것에 주의하자. 만약 그렇지 않다면 연결이 잘못 된 곳을 traceroute가 보일 수 없기 때문이다. 만약 traceroute가 제대로 연결된 것을 하나도 보이지 않았다면 그것은 당신의 테스트용 컴퓨터의 네트웍 설정 (default route, interface address, NIC drivers, DNS, etc.) 이 잘못되었다는 것을 나타낸다. 하나 정도의 연결이 보인다면 그것은 당신의 라우터가 ISP에 대해 제대로 설정되지 않았음을 의미할 것이다. 원하는 목적지까지 연결되지는 않았지만 몇 개의 경로를 지나갈 수 있었다면 그 문제는 대개 당신이 당장 손 댈 수 없는 외부의 ISP의 문제일 것이다.

4.4 유동 IP 사용하기

고정 IP 블록을 갖추고 다른 호스트의 서비스와의 상호 연결의 즐거움은 비용을 수반한다. 그것은 DSL을 이용한 고속 연결 혹은 케이블 모뎀 연결의 10회 정도의 연결과 맞먹는 비용이 된다. 만약 예산 문제로 그러한 연결이 불가능하거나 혹은 당신의 지역에서 그러한 연결이 지원이 되지 않는다면, 당신은 유동 IP를 사용할 것을 생각할 수 있을 것이다. 당신은 일반적으로 바로 그러한 것들 중 하나를 얻고 있다. 무슨 말이냐 하면 당신의 사설 네트웍 게이트웨이 머신은 외부에서 서비스가 들어올 때 그렇게 할 것이라는 말이다.

먼저 첫 번째로, 당신은 그것의 적법성을 확인해야만 한다. 많은 회사의 사용자 계약은 개인 계정으로 외부에서 접속할 수 있는 서버를 만드는 것을 금하고 있다. 그것은 내부로 들어오는 http 혹은 FTP 포트의 패킷 필터링을 실시하기 때문이다. 당신은 개인 계정의 다운링크, 업링크 속도가 가정용 DSL이나 케이블 모뎀에 비해 느리다는 것도 깨닫게 될 것이다. 업링크 속도는 FTP 서버나 웹서버에 달렸다.

당신이 유동 IP 를 사용할 것이라면, 그리고 연결이 들어오기를 바란다면, 당신은 유동 IP를 호스팅하는 DynIP *www.dynip.com/* `<http://www.dynip.com/>`, DynHOST *www.dynhost.com/* `<http://www.dynhost.com/>`, 혹은 TZO *www.tzo.com/* `<http://www.tzo.com/>`곳과 계약하게 될 것이다. 이와 같은 것들은 당신의 컴퓨터에 접속 프로그램으로서 작동하여 회사의 서버에 접속하게 한다. 당신이 현재 접속한 IP번호는 서버에서 살아 DNS 테이블에 새로운 값으로서 업데이트 된다. 당신은 그들의 도메인 이름에서 도메인 이름을 얻어 낼 수 있다. 예를 들면 “example.dynip.com” 이라던가 “example.dynhost.com”과 같은 혹은 당신 소유의 도메인을 1차적은 DNS 당국에 요청하여 공급받을 수도 있다. 대개는 비싸서 탈이지만.

Domain Host Services*www.dhs.org/* `<http://www.dhs.org/>` 와 같은 무료 호스팅 서비스에서도 마찬가지이다. 그것들은 새롭게 보이며 웹 사이트를 운영하기 쉽지만 당신은 별로 좋아보이지는 않는 곳을 찾을 것이다.

만약 당신이 유동 IP를 설정하기 위해 그런 업체들 중 하나와 계약을 하게 된다면, 그것은 도메인 서비스 업체를 선택하는 섹션이 당신의 결정중 하나에 작용한 것이다. 당신이 웹사이트 혹은 FTP 사이트응 운영하기 위해 호스팅을 할 예정이 아니었다면, 유동 IP서비스를 받기로 결정한 것은 큰 일도 아니다. 당신은

당신이 선택한 업체의 1차적인 DNS 권한을 설정하게 될 것이다. 당신은 외부에서 당신의 사설 네트웍으로 들어오는 데의 답변을 하는 이름붙은 데몬을 갖지 못할 것이다. 다른 세부적인 측면으로, E-mail을 잡을 때 당신이 계약한 곳의 서비스에 의존할 수 밖에 없을 것이다. 가장 좋은 방법은 그 회사의 스탭들의 도움을 받는 것이다.

꼬랑지 : 당신이 외부에서 유동 IP로의 원격지 연결을 원하지만 호스팅 서비스 는 사용하기를 원하지 않는다면, 외부 접속이 가능한 고정 IP가 지정된 컴퓨터에 "drop box"를 생성하고 당신의 유동 IP 호스트를 그것의 IP로 보내는 방식을 통해 저렴한 방식으로 email을 보내고 계정에 접속하여 파일에 접근할 수 있다. 당신이 원격지에서 당신의 컴퓨터에 접근하기를 원할 때 먼저 현재의 IP 주소를 drop box에서 끌어내고는 바로 그 IP주소로 slogin 을 사용하여 즉각 접근할 수 있다. 이것은 결국 유동 IP 호스팅 업체가 당신의 수고를 절약하기 위해 자동적으로 기본적으로 실시하는 일은 다 할 수 있다고 봐도 된다.

제 5 절 도메인 등록

어떤 사람이 외부에서도 자신이 선택한 도메인 네임을 갖는 웹서버 혹은 FTP나 e-mail 서버에 접속할 수 있기를 바란다면, 당신은 적절한 최상위 레벨의 도메인 데이터베이스에 그 도메인 이름을 등록해야만 할 것이다.

당신의 도메인 이름을 선택함에 있어 신중할 것을 당부한다. 일반적인 단어 혹은 어구는 대부분 일반적인 커뮤티니에서 선점하였을 것이고, 또한 당신의 출신과 언어가 다른 방문자를 불쾌하게 하는 속어의 사용은 안 될 것이다. 도메인 이름은 알파벳 26자와(악센트는 제외하고) 하이픈(시작이나 끝에는 둘 수 없다.) 그리고 10개의 숫자들만을 사용할 수 있다. 도메인 이름은 분류가 되지 않으며 최소 26자의 길이도 가능하다. 이미 존재하는 회사의 트레이드 마크 등의 알려진 것들을 등록하여 위법 행위를 저지르지 말도록 하라. 당신의 잘못 선택한 도메인으로 인한 형편에 대한 어떤 조언이라면, 당신의 관리에서 Uniform Domain Name Dispute Resolution Policy *www.icann.org/udrp/udrp-policy-24oct99.htm* `<http://www.icann.org/udrp/udrp-policy-24oct99.htm>`을 이용하라는 것이다.

많은 사람들이 이미 ".com", ".net", and ".org" 로 끝나는 최상위 레벨의 도메인을 등록했다. 당신의 도메인을 등록하기 전에 미리 다음의 곳 에서 그것이 이미 등록되었는지 알아 보는 편이 좋다. *www.icann.org/registrars/accredited-list.html* `<http://www.icann.org/registrars/accredited-list.html>`.

".ca", ".de", ".uk" 등등의 국명과 관련된 최상위 도메인을 등록하려면, 각 국가 최상위 도메인의 데이터베이스에서 제공하는 적당한 약관을 확인하라. *www.iana.org/cctld.html* `<http://www.iana.org/cctld.html>`.

대체로 당신은 첫번째나 두번째로 중요한 IP에 대한 접속 정보에 대한 기록을 제공받고, 의뢰의 수정계획(다른 누군가가 멋대로 당신의 도메인 정보를 바꾸기를 바라지는 않을 것이다.)을 비준하고, 연간 요금을 지불하게 될 것이다. 당신이 기록에 대한 의뢰 계획을 수정하는 것이 불편하게 느껴진다면, 그들에게 당신은 그들이 당신의 보안에 대한 관심을 제언하기 전에는 자발적으로 그 서비스를 이용하지 않을 것임을 알게 하라.

제 6 절 도메인 서비스 업체를 결정하기

거의 모든 서비스를 제공하는 ISP는 그들의 고객들을 위해 도메인 서비스를 제공한다. 이것은 어느 정도의 다른 사람들, 더욱 일반적인 데스크탑 혹은 서버 운영 시스템등에 호스팅과 더불어 서비스를 제공함으로서 더욱 풍부한 서비스를 한다. 이와 같은 서비스들은 리눅스 기반의 공급자에게 더욱 쉬운 것이고, 그런대로 저렴한 가격으로 하드웨어 호스팅을 가능하게 한다. 그러므로 당신은 당신 자신에게 꼭 맞는 서비스를 선택해야만 한다. 이런 서비스들은 대개 다음을 포함한다.:

? 당신의 도메인에 대한 1차적인 DNS 권한에 관해서는 "1차적인 DNS 권한" 을 보라.

? 전자 우편에 관해서는 "E-Mail"부분을 보라.

? 홈페이지를 올릴 공간을 호스팅하는 문제에 관해서는 “웹사이트 호스팅” 을 보라.

? FTP 사이트를 운영할 공간을 호스팅할 계획이라면 “FTP 사이트 호스팅”을 보라.

? 패킷 필터링에 관한 것은 “패킷 필터링”을 보라.

이것들의 각각의 사항에서 당신은 제어에 있어 기본적으로 편이성에 중점을 두게 될 것이다. 당신의 ISP에서 이들 중 하나 이상의 서비스를 제공할 때 당신은 그 서비스를 이용하는 다른 사람들의 경우를 보고 그런대로 확실히 생각하게 될 것이다. 그러므로 더 배워야 하거나 걱정을 할 필요까지는 없을 것이다. 같은 시간에 당신은 이런 서비스를 이용하는 데 있어서 제어를 하지 못할 수도 있을 것이다. 모든 변화는 당신이 원한다면 때때로 불편하거나 혹은 지연시키는 것들일 수도 있는 당신이 이용하는 ISP의 기술적인 지원을 포함한다. 이것은 보안상의 이슈를 포함하는 것이다. ISP는 공격자에게 개인의 것 보다 훨씬 더 매혹적인 타겟이 된다. ISP 서버가 e-mail과 홈페이지 공간을 고객들 게에 호스팅하면서부터 하나 이상의 서버의 명예를 이미 훼손시키고 그의 수고보다 많은 것을 얻어낸 하나 이상의 공격자(크래커 등등....)가 공격을 해올 때 데이터를 안전하게 보호하는 일 등이 그런 보안상의 이슈라 할 것이다.

6.1 1차적인 DNS 권한

외부인이 새로운 도메인인 example.com에 접속하려 할 때 명령들은 인터넷에 연결된 여러 서버를 지나 그 IP를 가진 기계를 마지막으로 찾고는 접속을 시도한 사람의 소프트웨어로 돌아가게 된다. 이와 같은 과정의 자세한 것은 이 문서의 성격을 넘어선 것이다. 많은 부분을 생략하고 이 요청이 fred.example.com에서 날아온 것일 때, 중추적인 데이터 베이스는 이 IP주소가 example.com에 대하여 1차적인 DNS에 수용되어 있는지를 문의한다. 이 IP 주소는 fred.example.com의 것을 질문하게 된다.

이것은 모든 도메인에 관한 1차적 혹은 2차적인 DNS서버에 관한 것이다. 중추 데이터 베이스에 기억되어 있는 이름과 IP 주소는 네트웍 솔루션 *www.networksolutions.com/* `<http://www.networksolutions.com/>`의 도메인 등록 권한에 의해 관리된다.

만약 당신이 당신의 ISP에 의해 호스팅된 1차 DNS 권한을 선택한다면, 이것들은 ISP를 관리하는 2개의 서버에 의한 것이다. 당신이 원하는 언제 라도 당신의 네트웍에 새로운 외부로 노출되는 컴퓨터를 추가할 수 있다. 당신은 ISP에 접근하여 새로운 컴퓨터와 그에 대한 데이터 베이스를 추가할 것을 요구할 수 있다.

당신이 당신의 도메인에 관한 1차적인 DNS 권한을 선택할 때 당신은 당신의 보조적인 다른 컴퓨터를 이용할 것이다. 전문적으로 당신은 과다한 인터넷 접속을 이용한다는 것이다. 그러나 보조적인 것을 당신의 ISP 에 연결하는 것은 아주 보편적이다. 당신이 당신의 네트웍에 외부로 보여지는 컴퓨터를 추가하기를 원한다면, 당신은 당신의 데이터 베이스를 업데이트 해야 하고, 약간의 시간 동안 프로퍼게이트의 갱신을 기다려야 할 것이다. 이와 같은 일들은 당신이 당신의 ISP를 거치지 않고 barney.example.com을 추가하기 위한 일이다.

보조 DNS를 호스트와의 사이에 두는 것은 좋은 생각이라 할 수 있다. 왜냐 하면, 당신과 ISP를 연결하는 단선의 케이블이 끊어질 경우 당신의 1차와 보조 DNS를 오프라인 상태로 만든다. 당신이 당신의 도메인을 등록하기 위해 이용한 도메인 등록자는 보조 DNS 서비스를 이용하게 해야 한다. 다음은 Granite Canyon*www.granitecanyon.com/* `<http://www.granitecanyon.com/>`의 무료 서비스로 누구이건 문의가 가능하다.

부주의혹은 당신이 당신의 도메인에 대한 1차 DNS의 권한을 선택하지 않음으로 인해 설정의도움이 필요할 때는 “네임 리솔루션의 셋업”을 보도록 하라. 당신은 당신의 사설 네트웍에서의 네임 리솔루션의 정리를 원하게 될 것이다. 그것은 당신이 1차 DNS 권한을 ISP에 위임했을 때의 현상이다.

6.2 E-Mail

당신이 당신의 ISP와 계약하였을 때 그들은 당신에게 사용할 수 있는 몇 개의 메일 주소를 주었을 것이다. 당신은 다음과 같은 경우에 이런 서비스를 독점적으로 이용하도록 할 수 있다. 예를들면 메일이 많이 와서 ISP서버의 당신의 메일 계정의용량이 차고 넘친다거나, 당신의 사용자들이 자신의 메일을 POP3 클라이언트를 이용하여 ISP의 서버에 연결하여 읽어야만 한다던가 하는 경우를 말한다. 대신하여 당신은 당신 자신

의 도메인으로 메일 주소를 갖는 메일을 당신의 서버에 설정할 수 있다. 다시 말해서 당신은 2가지의 접근을 통해 이점들을 얻을 수 있고, 당신이 좋아하는 것 한 가지를 택할 수 있다는 것이다.

당신이 메일을 이용하기 위해 ISP를 이용한다면 생각하라!:

그것은 집이나 혹은 비지니스를 위해 이동해 간 다른 장소에서 메일을 읽기 위해 접속하는 것이 더 편할 것이다. 또한 당신의 도메인을 보호하기 위한 보안의 안정성이 있다.

메일은 ISP의 서버에 기계적으로 보관되므로 원본의 디코딩에 있어 문제 를 일으킬 수도 있다.

당신은 메일 계정숫자에 제한을 갖게 되며, 계정을 늘리기를 원할 경우 돈을 지불해야 할 경우도 있다.

새로운 메일 주소를 필요로 할 경우, ISP를 방문해야 한다.

당신의 메일을 스스로 운영하겠다면 다음을 기억하라:

메일은 당신의 서버에 자동적으로 축적되고 당신의 ISP로 날아온 메일 들을 백업하여 보관할 경우 당신의 디스크는 꽉 차 버릴 수도 있다.

당신은 메일 주소를 몇 개를 만들거나 지우거나 해도 제한을 받지 않을 것이다.

당신은 당신의 네트웍에서의 메일 클라이언트를 지원해야 하고, 가능하면 메일 주소를 사용하는 사람들이 집에서도 자신들에게 날아온 메일들을 읽을 수 있도록 해야 한다.

하나의 가능한 접근 방법은 스스로 메일을 호스트하는 것이지만, 몇몇의 메일 주소를 ISP를 통해 관리할 수도 있을 것이다. 외부 네트웍에서 접속 가능한 메일 계정이 필요한 사람들은 자신의 ISP 메일 계정으로 포워딩되는 당신의 도메인 하의 메일 주소를 갖고 있을 수 있다. 다른 사람들은 사설 네트웍상의 지역 메일 주소를 갖고 있을 수도 있다. 이와 같은 것은 약간의 조정과 설정을 포함하지만 훨씬 유용성이 있는 방법이기도 하다.

당신이 당신의 도메인으로 된 메일 주소를 갖기 원한다면, “E-Mail 설정”의 설정 도움말을 읽기 바란다.

만약 당신이 당신의 도메인으로 된 메일 주소까지는 필요하지 않다면 “E-Mail 설정을 하지 않을 때의 DNS 설정”의 중요한 주석들을 참고하기 바란다.

6.3 웹사이트 호스팅

당신이 이용하는 ISP는 자신들의 웹 서버에 공간을 할당하여 당신에게 제공할 것이다. 당신은 그 공간을 이용하여도 좋고 혹은 당신의 외부로 공개되어 있는 네트웍 상의 컴퓨터를 이용하여도 좋을 것이다.

당신이 ISP의 웹 계정공간을 사용할 것이라면 다음을 기억하라:

당신은 당신이 주어진 용량을 초과하지만 않는다면 따로이 디스크 공간을 만들지 않아도 된다. 이것은 웹 내용의 공간 뿐 아니라 사람들이 사이트에 방문한 로그를 남기는 것을 모두 포함한다.

당신의 웹서버가 외부로 나가는 대역폭은 당신의 컴퓨터에 웹서버를 마련 한 것 보다 클 것이다. 이와 같은 경우 때문에 좀 느린 면이 있다.

CGI 스크립트나 기타 등등 당신의 웹 사이트에 필요한 프로그램들을 설치하는 것은 어려운 일에 속한다. 당신의 네트웍과 웹 서버 사이의 대역폭은 당신 자신의 네트웍에 설치한 경우보다 느릴 것이다.

당신 자신의 웹 공간을 이용하기로 했다면 기억하라:

당신은 호스팅 머신에 대해 더 많은 제어를 해야 할 것이다. 당신은 바로 당신의 어플리케이션을 위한 당신의 보안을 관리할 수 있다.

어쩌면 민감한 데이터, 다시 말해 신용 카드 번호라던가 메일 주소 등의 신상 정보들이 유출될 지 모른다.

당신의 백업 기술은 아무래도 ISP에서 해주는 것 보다는 떨어질 것이다.

나는 ISP가 데이터 전송 등에 있어 더욱 강력한 하드웨어를 보유하고 있다는 점 혹은 그와 같은 것들에 관해서는 그리 신경쓰는 편이 아니다. 하지만 당신이 고사양의 네트웍 상에서의 데이터 전송을 필요로 한다

면, 이것은 중요한 문제가 된다. 그리고 솔직하게 말하자면, 이에 대한 판단은 리눅스 하우투를 보는 것 보다는 실력있는 컨설턴트에게 위임을 해 두는 편이 낫다고 본다.

당신은 당신의 서버에 당신이 도메인으로 나가는 웹 공간과 설정을 위한 다른 인터넷 하우투 문서 등이 필요하다고 생각되는지. 나는 이와 같은 것이 사설 네트웍 게이트웨이의 서비스와는 다르다는 것을 보안상의 이유 때문에라도 말해 두도록 하겠다.

6.4 FTP 사이트 호스팅

기본적으로 FTP 호스팅과 WWW 호스팅은 FTP가 CGI 스크립트 등의 동적인 문서를 필요로 하지 않는다는 것을 제외하고는 동일한 개념인 것이다. 근래의 ftpd 개발 상황은 아무나 와서 자료를 올릴 수 있는 업로드 디렉토리에 너무 긴 이름의 디렉토리를 생성하는 것과 같은 버퍼 오버런의 해결이 이루어진 것이 가장 중요한 일이 될 것이다. 당신의 ISP가 업로드를 용인하는 것은 물론이고 FTP 데몬에 관한 보안에 있어 느슨한 태도를 보인다면, 당신은 그 서비스를 해지하는 편이 좋을 것이라고 본다.

당신이 당신의 서버에 직접 당신의 도메인 이름을 가진 FTP 호스팅을 하기 원한다면, 최근 버전의 FTP 데몬을 구하도록 하고 그에 대한 명령 설정을 고려하라. 한가지 더, 나는 이와 같은 서비스를 하는 것이 사설 네트웍 안에서의 것과는 다르다는 것을 말해 두고 싶다. 보안에 주의하라.

wu-ftpd를 위해 나는 일반적인 설정 옵션을 기술해 두도록 하겠다.:

? –disable-upload - 익명의 업로드 서비스를 필요로 하지 않을 때.

? –enable-anononly - 당신의 사용자들이 컴퓨터 간의 파일 이동을 위해 scp를 사용하는것을 장려할 때.

? –enable-paranoid - 현재 버전에서의 특집 격의 무엇이건 의심스럽게 고려하는 일은 없게 한다.

6.5 패킷 필터링

어떤 ISP 는 내부 혹은 외부에서의 공격자들을에 대비하여 그들의 네트웍으로 전송되는 패킷을 필터링한다. 케이블 모뎀 네트웍 혹은 비슷한 양식의 네트웍은 윈도우 98 혹은 95의 디스크 공유를 통해 그들의 하드드라이브에 있는 내용이 외부의 네트웍에 연결된 누구에게나 노출되는 난처한 상황을 겪기도 한다. 이와 같은 케이스들에서 해결책이라면 사용자들에게 그러면 안 됨을 말하는 것이겠지만 어떤 공급자들은 하드웨어에 접속하는 사람들을 필터링하여 그 문제 를 해결하고 있다.

패킷 필터링은 당신 자신을 위해서 하는 편이 나은 일이라 할 수 있다. 그것은 당신의 사설 네트웍 게이트웨이 컴퓨터에서 작동하는 커널에 적합하고 당신 에게 무슨 일이 일어났을때 좀 더 효과적인 대처를 하게 해 준다. 당신은 종종 당신이 방화벽의 효과적인 셋업을 위한 묘안이 필요하다는 것을 느낄 것이다. 그리고 그것은 실시간 상에서의 기술적인 보조보다 쉬운 일이다.

당신이 패킷 필터링을 당신의 도메인에서 하기로 결정하였다면, "패킷 필터링의 설정"의 섹션을 설정상의 도움을 위해 보도록 하라.

제 7 절 서비스의 설정

7.1 네임 리솔루션의 셋업

당신은 당신의 네트웍에 연결된 컴퓨터가 이름만으로도 나타낼 수 있기를 바랄 것이다. 그리고 외부의 사람 역시 ip 주소를 치고 들어가는 것 보다는 이름으로 연결하기를 바랄 것이다. 다음은 그리 하기 위한 몇몇 가지 방법이다.

7.1.1 사설 네트웍 상에서 도메인을 지휘하는 DNS, ISP

[참고 : 만약 당신이 사설 네트웍을 충족시키지 않는 것을 선택해야 한다면 "ISP에 의해 호스팅 되는 노출된 네트웍"의 섹션을 참고하라.]

이 설정에서 당신은 당신의 도메인에 대한 1차 DNS의 권한을 위임하게 된다. 당신은 당신의 사설 네트웍에서 컴퓨터가 다른 컴퓨터와 talk 를 하기를 원할 때에나 DNS 를 이용하고 있다. 당신은 당신의 ISP에 당신의 외부로 노출시키기를 원하는 컴퓨터들의 IP주소 목록을 알려 주어야 한다. 만약 당신이 betty.example.com 이라는 컴퓨터를 웹과 FTP 서버로서 외부에 노출시키기를 원한다면, ISP에 www.example.com과 ftp.example.com을 betty.example.com으로 연결하도록 해 줄 CNAME 목록을 작성해 줄 것을 요청하라.

당신의 사설 네트웍 게이트웨이 머신 상의 DNS 을 설정하라. 이것은 보안상의 문제이며 또한 당신의 DNS 접근 문제에 대한 해결로서 업그레이드를 용이하게 한다.

나는 당신이 사설 네트웍 게이트웨이의 dns.example.com라는 기계를 DNS로 사용함을 가정하고 이것이 192.168.2.1의 IP를 가진 fred.example.com 으로 알리아스 됨을 가정하겠다. 상황에 따른 약간의 수정이 필요하다. 나는 이 HOWTO에서 그런 점까지 다룰 것은 아니라고 생각한다.

당신은 BIND 즉 버클리 인터넷 네임 도메인의 최근의 버전을 다운로드 받아 야 한다. 그것은 BIND 웹 사이트*www.isc.org/products/BIND/* `<http://www.isc.org/products/BIND/>`에서 찾아 볼 수 있다. 다음으로 당신은 데몬을 설정해야 한다. 다음의 /etc/named.conf 라는 파일을 생성하여라.

```
options {
            directory "/var/named";
            listen-on { 192.168.1.1 };
    };

    zone "." {
            type hint;
            file "root.hints";
    };

    zone "0.0.127.in-addr.arpa" {
            type master;
            file "pz/127.0.0";
    };


    zone "1.168.192.in-addr.arpa" {
            type master;
            file "pz/1.168.192";
    };

    zone "example.com" {
            type master;
            notify no;
            file "pz/example.com";
    };
```

우리가 example.com 도메인의 주인임을 선언했다는 것을 기억하라. 이것은 우리의 ISP 역시 어떤 도메인의 주인임을 선언하고 있다는 것과 같다. 이것은 당신이 셋업에 주의하는 한 전혀 문제가 되지 않는다. 사

설 네트웍 상의 모든 컴퓨터들은 dns.example.com을 네임 리솔루션으로서 사용한다. 그들은 ISP의 네임 서버가 당신이 등록한 도메인들에 대하여 신뢰도를 유지하는 한 ISP상의 네임 서버를 사용할 필요가 없는 것이다. 그러나 ISP의 네임 서버는 당신의 사설 네트웍 상의 컴퓨터들이 가진 IP번호에 관해서는 아는 바가 없다. 간단히 말하자면 ISP의 네임 서버가 아는것은 당신의 외부로 노출되어 있는 컴퓨터의 IP 번호인 것이다. dns.example.com 아래에 있는 사설 네트웍에 관한 것이 아니라.

/var/named 밑에 변수 파일을 생성하라.

root.hints 파일은 BIND 문서에서 기술하는 것이다. 혹은 DNS HOWTO *metalab.unc.edu/pub/Linux/docs/HOWTO/DNS-HOWTO* `<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/DNS-HOWTO>`에서도 유효한 roo.hints 파일을 만드는것에 관해 자세히 나와 있다.

```
H.ROOT-SERVERS.NET.     6d15h26m24s IN A  128.63.2.53
C.ROOT-SERVERS.NET.     6d15h26m24s IN A  192.33.4.12
G.ROOT-SERVERS.NET.     6d15h26m24s IN A  192.112.36.4
F.ROOT-SERVERS.NET.     6d15h26m24s IN A  192.5.5.241
B.ROOT-SERVERS.NET.     6d15h26m24s IN A  128.9.0.107
J.ROOT-SERVERS.NET.     6d15h26m24s IN A  198.41.0.10
K.ROOT-SERVERS.NET.     6d15h26m24s IN A  193.0.14.129
L.ROOT-SERVERS.NET.     6d15h26m24s IN A  198.32.64.12
M.ROOT-SERVERS.NET.     6d15h26m24s IN A  202.12.27.33
I.ROOT-SERVERS.NET.     6d15h26m24s IN A  192.36.148.17
E.ROOT-SERVERS.NET.     6d15h26m24s IN A  192.203.230.10
D.ROOT-SERVERS.NET.     6d15h26m24s IN A  128.8.10.90
A.ROOT-SERVERS.NET.     6d15h26m24s IN A  198.41.0.4
```

pz/127.0.0 파일은 다음과 같다.:

```
$TTL 86400

@               IN      SOA     example.com. root.example.com. (
                                1       ; Serial
                                8H      ; Refresh
                                2H      ; Retry
                                1W      ; Expire
                                1D)     ; Minimum TTL
                        NS      dns.example.com.
1                       PTR     localhost.
```

pz/1.168.192 파일은 다음과 같다:

```
$TTL 86400

@       IN      SOA             dns.example.com. root.dns.example.com. (
                                1       ; Serial
                                8H      ; Refresh 8 hours
                                2H      ; Retry   2 hours
                                1W      ; Expire  1 week
                                1D      ; Minimum 1 day
                        )
                NS      dns.example.com.
```

```
1               PTR     fred.example.com.
                PTR     dns.example.com.
                PTR     mail.example.com.
2               PTR     barney.example.com.
3               PTR     wilma.example.com.
```

그리고 당신이 사설 네트웍의 각각의 컴퓨터에 생성하는 PTR 레코드들이 있다. 이 예에서 fred.example.com의 IP 주소는 192.168.1.1이고 이것은 dns.example.com과 mail.example.com으로 알리아스 된다. 192.168.1.2의 IP 주소를 가진것은 barney.example.com이고 다른 것들도 마찬가지이다.

pz/example.com 파일은 다음과 같다.:

```
$TTL 86400

@               IN      SOA     example.com. root.dns.example.com. (
                                1       ; Serial
                                8H      ; Refresh 8 hours
                                2H      ; Retry   2 hours
                                1W      ; Expire  1 week
                                1D      ; Minimum 1 day
                        )
                        NS              dns.example.com.
        IN              A               192.168.1.1
        IN              MX          10  mail.example.com.
        IN              MX          20  <ISP mail machine IP>.


localhost               A           127.0.0.1
fred                    A           192.168.1.1
                        A           10.1.1.9
dns                     CNAME       fred
mail                    CNAME       fred
barney                  A           192.168.1.2
wilma                   A           192.168.1.3
betty                   A           10.1.1.10
www                     CNAME       betty
ftp                     CNAME       betty
```

우리는 사설 네트웍상의 컴퓨터들과 외부로 나가는 IP들에 대한 목록을 모두 생성했다는 것을 기억하라. 이것은 당신의 ISP 네임 서버에는 당신의 사설 네트웍상의 컴퓨터에 대한 기록이 없기 때문이다. 이것은 다시 말해서 사설 네트웍상의 betty.example.com이 외부로 노출된 fred와 같은 IP를 사용할 수도 있다는 것을 의미한다.

/etc/named.conf의 한 줄에 대해서는 옵션 섹션을 참고하라.:

```
listen-on { 192.168.1.2 };
```

이것은 당신의 네임 데몬이 외부에서의 DNS 질의에 대답하는것을 예방하는 것이다.(외부에서의 모든 질의는 ISP의 네임 리졸버가 해결할 문제인 것이다.)

7.1.2 DNS 리솔루션이 필요없는 사설 네트웍상에서의 도메인 설정

[참고: 만약 당신이 사설 네트웍을 사용하지 않을 것이라면, "ISP에 호스팅 되는 노출된 네트웍" 섹션을 참조하라.] 이와 같은 설정에서 당신은 당신의 사설 네트웍을 웬만큼 작게 그리고 수시로 바꿀 수 있음을 결정하는 것이다. 당신은 DNS의 중추적인 데이터 베이스를 사용하지 않음을 결심해야 하는것은 물론이다. 그리고 각각의 컴퓨터에 대한 리솔루션 역시 기대하지말기 바란다. 이 사설 네트웍에 연결된 모든 컴퓨터들은 그 네트웍의 게이트웨이가 되는 그들의 호스트 네임을 DNS에서 인식하는것이 전부인 것이다. 사설 네트웍에 있어서의 네임 리솔루션에 있어서 호스트 테이블이 생성되어야만 한다. 리눅스에서는 이것은 사설 네트웍 상의 각각의 컴퓨터의 /etc/hosts 파일 안에 각각의 컴퓨터의 IP 주소와 이름이 기록되어 있어야 한다는 뜻이다. 언제라도 새로운 컴퓨터를 추가할 수 있고 이름이나 IP 주소를 멋대로 바꿀 수도 있는 이 파일은 리눅스 컴퓨터에 의해 업데이트 된다.

이 섹션에서 외부로 나가는 IP번호를가진 컴퓨터의 목록은 반드시 ISP로 보내어진다. 그리고 모든 알리아스(www와 ftp 등에서의) 상에서는 CNAME 을 ISP에 생성하는 특별한 방법을 이용하게 된다.

7.1.3 당신이 도메인에 대한 1차 DNS 권한을 갖고 있을때

당신이 외부로 노출된 네트웍을 위해 그리고 사설 네트웍을 위해 네임 리솔루션을 설정할 수 있는 상황이라면 이 글은 필요가 없을 것이다. 만약 당신이 한 대의 컴퓨터를 이용하여 두 가지 일을 모두 해결하고 싶다면 그것은 간단한 설정으로 해결될 수 있다. 이 섹션에서는 사설 네트웍 게이트웨이 머신으로 내부와 외부 모두의 요청을 처리할 수 있는 방법을 말하겠다.

이 글을 쓰고 있는 시점인 BIND 8.2.2가 나와 있는 상황에서 하나의 데몬을 요청이 도착하는 상황에서 서로 다른 설정에 의한 요청에의 답변을 하게 할 수는 없다. 우리는 네임 리솔루션이 내부의 네트웍의 IP 번호가 밖으로 노출되지 않았기 때문에 생기는 외부의 요청과, 그리고 대답해야만 하는 내부의 요청이라는 서로 다른 두 가지 요구를 모두 처리하기를 원한다. 이것은 BIND의 미래 버전에서 "view"라는 키워드를 통해 발견할 수 있게 될 것이다. 그러나 아직까지는 서로 다른 설정의 두 가지의 데몬을 띄우는 방법을 통해 해결하게 될 것이다.

첫 번째로, "사설 네트웍 상에서 도메인을 지휘하는 DNS, ISP " 섹션 을 참고하여 사설 네트웍의 도메인을 설정하라. 이것은 내부 네트웍에서 네임 리솔루션을 수행하게 한다.

다음으로, 당신의 외부로 노출되어 있는 도메인을 위한 DNS를 설정하라. 첫 번째로, 당신의 인터넷 공급자에게 당신의 IP를 대리로 처리하게 하라. 원래의 표준적인 DNS가 C클래스이하의 서브넷에서의 역변환 조정을 지원 하지 않는 동안, 주의는 DNS 클라이언트의 불만에 대한 일들을 발전시키고, RFC2317의 개요를 따른다. *www.ietf.org/rfc/rfc2317.txt* `<http://www.ietf.org/rfc/rfc2317.txt>`만약 당신의 공급자가 당신의 IP블록에 대한 조정을 할 수 있다면, 당신은 그들에게 그들이 위임할 -RFC는 그들이 매일 사용할 수 있게 되는 것에 대한 약정을 제시하는데 대한 제언을 하지 않았다.-가상 도메인을 요구할 것을 결정해야 한다. 나는 공급자가 당신에게 위임받은 상태임을 가정하고 가상의 도메인은 8.1.1.10.in-addr.arpa.라고 해 두겠다. 공급자는 CNAME 목록을 다음과 같은 형식으로 생성할 것이다.

```
8.1.1.10.in-addr.arpa.     2H IN CNAME 8.8.1.1.10.in-addr.arpa.
9.1.1.10.in-addr.arpa.     2H IN CNAME 9.8.1.1.10.in-addr.arpa.
10.1.1.10.in-addr.arpa.    2H IN CNAME 10.8.1.1.10.in-addr.arpa.
etc.
```

1.1.10.in-addr.arpa 도메인을 위한 이와 같은 파일에 말이다. 이와 같은 설정에 관해서는 다음 섹션에서 거론될 것이다.

만약 당신의 공급자가 당신에게 역 DNS를 관리해 줄 수 있다면 그들은 당신의 보여지는 가상 도메인의 기록과 일치하는 것을 지정해 내어 당신이 관리하는 IP에 대한 역 DNS 범위의 테이블에 대해 CNAME 순서를 생성할 것이다. 만약 그들이 당신에게 그런 일을 해 주지 않는다면,당신은 그들에게 당신의 도메인 중에서 외부로 보여지는 것을 순서에 추가하거나, 지우거나, 이름을 바꾸는 것을 일일히 요청해야 한다. 역 DNS

테이블이 당신의 외부로 노출되어 있는 DNS 들과 동시에 일어나지 않는다면, 어떤 서비스들은 장애를 일으킬 것이고, 그렇지 않은 것들은 잘못된 처리를 하게 될 것이다.

당신은 이제 사설 네트웍의 게이트웨이이며 외부의 요구 상황에 응하도록 설정할 컴퓨터의 두번째 이름을 셋업해야 한다. 이 설정 리스트는 오직 외부로 노출되어 있는 컴퓨터들과 외부 요청을 받아 처리하는 사설 네트웍 게이트 웨이에 관한 호스트와 IP주소만을 요구한다.

먼저, 외부 요청에 대한 설정인 /etc/named.ext.conf라는 두 번째의 설정 파일을 생성하라. 우리의 예에서는 다음과 같이 나타난다.:

```
options {
        directory "/var/named";
        listen-on { 10.1.1.9; };
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "pz/127.0.0";
};


zone "8.1.1.10.in-addr.arpa" {
        type master;
        file "ext/8.1.1.10";
};

zone "example.com" {
        type master;
        notify no;
        file "ext/example.com";
};
```

/var/named 밑에 있는 root.hints 와 pz/127.0.0 파일은 실행되는 데몬을 공유한다. ext/8.1.1.10 파일은 다음과 같다.:

```
$TTL 86400

@       IN      SOA             fred.example.com. root.fred.example.com. (
                                1               ; Serial
                                10800           ; Refresh       3 hours
                                3600            ; Retry         1 hour
                                3600000         ; Expire        1000 hours
                                86400 )         ; Minimum       24 hours
                NS      dns.example.com.
9       IN      PTR     fred.example.com.
                PTR     dns.example.com.
```

```
                    PTR     mail.example.com.
10      IN          PTR     betty.example.com.
                    PTR     www.example.com.
                    PTR     ftp.example.com.
```

ext/example.com 파일은 다음과 같은 내용이다.:

```
$TTL 86400

@               IN      SOA     example.com. root.fred.example.com. (
                                10021   ; Serial
                                8H      ; Refresh 8 hours
                                2H      ; Retry   2 hours
                                1W      ; Expire  1 week
                                1D      ; Minimum 1 day
                        )
                        NS              fred.example.com.
        IN              A               209.217.100.58
        IN              MX          10  mail.example.com.
        IN              MX          20  <ISP Mail Machine>.


localhost               A           127.0.0.1
fred                    A           10.1.1.9
betty                   A           10.1.1.10
dns                     CNAME       fred
mail                    CNAME       fred
www                     CNAME       betty
ftp                     CNAME       betty
```

사설 네트웍 게이트 웨이에서 2개의 데몬을 기동시켜라. 다음을 당신의 네트웍 데몬의 시동 스크립트에 추가하라.

```
/usr/sbin/named -u dnsuser -g dnsgroup /etc/named.conf
/usr/sbin/named -u dnsuser -g dnsgroup /etc/named.ext.conf
```

나는 당신이 별 특권이 없는 유저인 “dnsuser”와 그가 소속되어 있는 특권 없는 그룹인 “dnsgroup”을 생성했다고 가정한다. 만약 이 안의, 외부의 공격자가 네임 데몬을 통해 들어올 버그가 있을 때, 공격자는 운영이 제한되는 별 특권 없는 유저의 권한을찾게 될 것이다. /var/named 디렉토리와 그 안의 파일들은 “dnsuser”와 같은 일반 유저에게 기록 권한을 주어서는 안 되는것이다.

사설 네트웍 상의 컴퓨터는 dns.example.com (우리의 예에서 192.168.1.1의 주소를 가진)로 설정된 네임리솔루션을 가진다. 또한 외부로 노출된 컴퓨터는 외부 인터페이스의 네트웍 게이트웨이 (우리의 예에서는 10.1.1.9 였다.)나 혹은 ISP의 DNS 서버에 질의할 수 있다.

7.1.4 ISP에 의해 호스팅 되는 노출된 네트웍

이 설정에서 당신은 당신의 호스트 아래 있는 모든 네트웍을 노출시킨다. 당신은 당신의 컴퓨터 각각에 할당할 당신의 ISP에 컴퓨터의 이름과 IP가 기록되어 있는 실재하는 IP주소를 가지고 있을 것이다. ISP는 당

신에게 최소 한 개 이상의 IP주소를 등재해 두었을 것이다. 당신의 리눅스 박스는 /etc/resolv.conf에서 네임 리솔루션을 위한 설정이 이루어져야 한다.:

```
search example.com
nameserver <DNS host 1>
nameserver <DNS host 2>
```

윈도가 깔려 있는 컴퓨터는 네트웍 설정에서 같은 방식으로 설정할 수 있다.

7.1.5 도메인을 옮기기 전의 DNS의 준비

당신이 당신의 도메인을 새로운 IP로 옮기기를 원한다면 당신은 그 전에 준비 사항으로서 몇 가지 할 일이 있다.

당신은 IP번호를 DNSlookup에서 가져와 다른 외부의 연결되는 고유의 IP로 옮기기를 원하고, 옮기자마자 빠르게 사용하기를 원할것이다. 원결의 사이트는 당신의 IP번호를 캐시에 저장하고 있을 것이므로 사용자의 요청에 의해 당신의 이전 IP번호를 제공할 수 있다. 이와 같은 현상은 최근에 당신의 사이트를 방문한 사람들이 연결을 회피하게 하는 원인이 된다. 새로운 방문자, 즉 캐시가 저장되어 있지 않은 사람들에게는 이와 같은 일은 문제가 되지 않는다. 일을 복잡하게 되도록 조장하는 것은 사실은 최상위 레벨의 서버가 하루에 두 번 밖에 갱신이 되지 않는 다는 것엔데다가 당신의 DNS서버가 최상위 레벨의 서버와 동시에 갱신되게 하기 위해 시간을 맞추는 것은 쉬운 일이 아니라는 점에 있다.

옮기는 작업에 있어 가장 쉬운 방법이라는 것은 거의 사이트를 복제하는 것이 가장 좋은 해법일 것이다. 혹은 눈에 보이는 부분만이라도. 새로은 IP번호로 바뀌어지는 일이 전달되고 그리고 잠시 기다리면 새로운 IP번호로 연결된다. 이것이 가장 어렵지 않은 방법이다.

당신이 당신의 새로운 ISP에 대하여(혹은 IP 번호만 새로운 것을 이용하고 기존의 ISP를 그대로 이용할 수도 물론 있다.) 처음으로 해야 하는 일은 1차 및 2차 DNS를 갱신하는 것이다. 이런 작업은 IP를 옮기기 하루 전에는 해 주는 것이 좋다. 그들에게 레코드 상에 적당히 작은 어떤 것인 TTL을 설정할 것을 주문하라. (5분이면 할 수 있을 것이다. 역자 주: 근데 TTL이 뭔지 잘 모르겠네요...... 대충 찍기로 이해해 보는 거죠, 뭐! ^^;;;) DNS의 샘플 파일은 TTL값을 86400 초 , 그러니까 하루로 설정하는 것을 빠르게 한다. 만약 당신이 이것보다 TTL값을 길게 잡기를 원한다면 당신은 빠른 시일 내에 다른 것들을 더 옮길 생각인 것이어야 옳다. 궁극적으로 이것은 당신의 성취이다. 만약 당신의 현재의 도메인 정보의 TTL이 N시간으로 되어 있다면, N시간이 다 지나기 전에 옮기는 작업을 하도록 하라.:

당신의 도메인 등록은 새로운 공급자의 1차와 2차 DNS의 루트 데이터 베이스에서 이루어 져야 한다. 이 작업을 하기 최소 하루 전에 데이터 베이스에 변경 사항과 시간을 알려야 한다.

새로운 1차와 DNS 서버는 아주 적은 TTL로 당신의 사이트의 실제의 IP를 찍어 낼 수 있어야 한다.

당신은 지정한 N시간 안에 작업을 다 마쳤다고 하더라도 당신의 현재 도메인의 TTL값을 가속시켜서는 안된다.

자, 당신은 이제 옮길 준비가 다 되었다. 당신의 컴퓨터를 새로운 IP주소로 갱신하라. 당신의 ISP의 DNS 레코드가 그와 함께 갱신된다. 5분 안에 (당신이 옮기기 위해 설정할 짧은 TTL) 연결이 새로운 사이트로 이루어진다. 당신은 DNS의 저작을 당신의 연결된, 당신이 원한다면 만들 수 있는 당신 자신의 서버에도 옮길 수 있지만 그러기 위해서는 진작에 TTL 값을 크게 잡았어야 했을 것이다.

7.2 E-Mail 호스팅을 하지 않을 때의 DNS설정

이 설정은 "네임 리솔루션의 셋업" 섹션의 MX 레코드 포인트인 "mail.example.com" 컴퓨터를 초점으로 하여 설명하다. 가장 작은 번호로서 우선권을 갖는 MX 레코드는 메일을 보낼 원격 사이트를 받는다. 뒷순위를 갖는 다른 MX 레코드는 종종 받아진 메일을 백업한다. 이런 백업들은 때때로 1차의 메일을 받는 컴퓨터가 어떤 이유로 그 메시지를 받을 수 없을 경우의 시간의 경과에 메일들을 유지시킨다. 이 섹션의 주어지

는 예에서는 fred.example.com 이라는, mail.example.com의 앨리어스를 받는 도메인에 의한 E-Mail을 사용하는 컴퓨터를 가정하자. 당신이 당신의 메일 호스팅을 ISP를 사용할 것을 선택한다면, 당신은 MX 레코드를 ISP의 컴퓨터로 돌릴 수 있다. 당신의 ISP의 기술적인 지원으로 당신은 MX 레코드를 가지각색의 파일들 안에서 전형적으로 사용할 수 있다.

7.3 E-Mail 설정

당신이 당신의 도메인 하에 E-MAIL을 완전히 설정하여 호스팅 할 것이라면, 당신은 사설 네트웍 상에서 메일을 받아 네트웍상의 어느 컴퓨터 상에서도 읽을 수 있게 하기 위해 특별한 방법들을 사용하여야 한다. 당신의 주의 없이는 메시지들은 호스트에 아닌 컴퓨터에만 로그인 하는 그 메일의 수신자에게 오랫동안 전달되지 못한 채 호스트에서 대기할 지도 모른다. 보안의 차원에서, 나는 도착되는 메일이 외부의 눈에 띄이는 컴퓨터(그것들은 아마 실제 IP를 자신의 데스크탑 컴퓨터에서 사용하기를 원하는 PHB를 낙심시키는 대신 그가 하루에 두 번 죽음의 ping을 보내더라도 놀랄 수만은 없게 할 것이다.) 전송된 메일은 가장 깔끔한 방법인 semdmail에서의 포워딩을 통해 사설 네트웍 내에서만 공유되어야 한다. 어떤 다른 메일 핸들링 데몬을 위한 해법 테스트를 제공하기를 원하는 사람이 있다면 감사히 첨부하겠다.

7.3.1 "sendmail"을 사용한 해법

하나의 호스트에 도착한 메일을 다른 모든 컴퓨터에서 읽게 하는 가장 간단한 방법은 메일 스풀 디렉토리를 내부 네트웍 상에서 읽고 쓰기가 자유롭게 하는 것이다. 사설 네트웍의게이트웨이 컴퓨터는 사설 네트웍 전체를위해 메일을 보내고 받는 작업을 수행할 것이다. 그리고 루트는 메일 스풀 드라이브에 대한 쓰는 권한을 가질 수 있고 다른 클라이언트들은 루트가 금지하지 않는 이상은 재량껏 그런 권한을 누릴 수도 있을것이다. 나의 일반적인 보안 철학이라는것은 권한이라는 것은 되도록 주지 않는 데 있으므로 나는 사설 네트웍의 게이트웨이 머신이 받아들이는 메일 스풀이 되어 있는 네트웍 드라이브에 대한 권한을 제한하는 쪽이다. 이것은 루트는 오직 자신에게 도착한 메일들을 읽을 수 있을 뿐이지만 이것은 특별히 중요한 문제점은 아니다. 메일 스풀 드라이브는 NFS를 거쳐 사설 네트웍 게이트웨이 컴퓨터 안의 디렉토리로서 존재하거나 혹은 내부의 서버 상에 디렉토리로서 존재해야 한다. 만약 메일 스풀 드라이브가 사설 네트웍 게이트웨이 컴퓨터 상에 존재한다면 그 컴퓨터에 관한 루트의 제한이 발생 될 수 없다. 만일 이것이 다른 서버에 위치한다면, 메일이 서버나, 게이트웨이 컴퓨터나, 혹은 네트웍 연결로 넘겨지지 않아 다운될 것임을 기억하라.

사설 네트웍 상의 윈도 머신에 대해, 메일 스풀 호스트에 POP서버를 설치하거나 혹은 삼바를이용하여 메일 스풀을 보내 줄 수 있다. 윈도 머신은 메일을 리눅스 상의 유저로 받는 것과 되돌려 보내는 것에 관해 설정되어야 한다. 예를 들어 joeuser@example.com와 같은 유저의 메일 주소는 유저 네임과, barney.example.com와 같은 컴퓨터 이름이 아닌 도메인 네임으로 된 호스트 명으로 구성되어 있는 것이다. 외부로 나가는 SMTP 호스트는 내부에서의 주소를 반송받을 수 있는 주소로 재작성하여 내보내는 것이다.

다음으로 당신은 사설 네트웍에서 메일을 포워딩하고 메일 주소를 재작성하기 위해 센드메일을 설정해야 한다. 가장 최근의 센드메일 소스는 sendmail.org의 웹 사이트인 *www.sendmail.org/* `<http://www.sendmail.org/>`에서 얻을 수 있다. 컴파일 된 바이너리를 얻고 싶다면 센드메일 소스들이 있는곳의 cf/domain 디렉토리에서 받고, 이제 다음의 example.com.m4와 같은 새로운 파일을 생성하라.:

```
divert(-1)
#
# Copyright (c) 1998 Sendmail, Inc.  All rights reserved.
# Copyright (c) 1983 Eric P. Allman.  All rights reserved.
# Copyright (c) 1988, 1993
#       The Regents of the University of California.  All rights reserved.
#
# By using this file, you agree to the terms and conditions set
# forth in the LICENSE file which can be found at the top level of
```

```
# the sendmail distribution.
#
#

#
#  The following is a generic domain file.  You should be able to
#  use it anywhere.  If you want to customize it, copy it to a file
#  named with your domain and make the edits; then, copy the appropriate
#  .mc files and change ‘DOMAIN(generic)’ to reference your updated domain
#  files.
#
divert(0)
define(‘confFORWARD_PATH’, ‘$z/.forward.$w+$h:$z/.forward+$h:$z/.forward.$w:$z/.forward’)dnl
FEATURE(redirect)dnl
MASQUERADE_AS(example.com)dnl
FEATURE(masquerade_envelope)dnl
```

이것은 “example.com”의 도메인을 정의한다. 다음으로 당신은 메일 호스트로 사용되고(사설 네트웍 게이트웨이 머신) 사설 네트웍 상의 다른 리눅스 머신들과 연결되어 있는 곳에 sendmail.cf 파일을 생성해야 한다.

다음과 같은 파일을 센드메일 소스 트리 상에 cf/cf에, example.master.m4 로서 생성하라.:

```
divert(-1)
#
# Copyright (c) 1998 Sendmail, Inc.  All rights reserved.
# Copyright (c) 1983 Eric P. Allman.  All rights reserved.
# Copyright (c) 1988, 1993
#       The Regents of the University of California.  All rights reserved.
#
# By using this file, you agree to the terms and conditions set
# forth in the LICENSE file which can be found at the top level of
# the sendmail distribution.
#
#

#
#  This is the prototype file for a configuration that supports nothing
#  but basic SMTP connections via TCP.
#
#  You MUST change the ‘OSTYPE’ macro to specify the operating system
#  on which this will run; this will set the location of various
#  support files for your operating system environment.  You MAY
#  create a domain file in ../domain and reference it by adding a
#  ‘DOMAIN’ macro after the ‘OSTYPE’ macro.  I recommend that you
#  first copy this to another file name so that new sendmail releases
#  will not trash your changes.
#

divert(0)dnl
OSTYPE(linux)dnl
```

```
DOMAIN(example.com)dnl
FEATURE(nouucp)
FEATURE(relay_entire_domain)
FEATURE(‘virtusertable’, ‘hash /etc/sendmail/virtusertable’)dnl
FEATURE(‘genericstable’, ‘hash /etc/sendmail/genericstable’)dnl
define(‘confPRIVACY_FLAGS’, ‘‘noexpn,novrfy’’)dnl
MAILER(local)
MAILER(smtp)
Cw fred.example.com
Cw example.com
```

이 예에서 우리는 “expn” 과 “vrfy” 명령을 사용 불가능하게 해야 한다. 공격자는 “expn”을 이용하여 앨리어스에 대한 낚시질(적당한 단어가 생각이 나지않는군요.....)을 그가 외부의 그의 계정에서 공격을 할 때 까지 “staff”, “allstaff”, “office” 등의 이름으로 할 수 있기 때문이다. 그는 그가 얻어내기 쉬운 취약한 패스워드를 지닌 계정을 얻으려 할 것이다. (그의 앞에 로그인 화면이 뜰 때 를 가정하자 - 보안 설정의 설명은 “당신의 도메인에 대한 보안” 에서 외부의 공격자에게 로그인 프롬프트를 띄우지 않는것에 관해 말하며 하겠다.)

당신이 마저 생성해야 하는 예를 들어 example.slave.m4라는 종속된 컴퓨터를 위한 sendmail.cf는 다음과 같다.

```
divert(-1)
#
# Copyright (c) 1998 Sendmail, Inc.  All rights reserved.
# Copyright (c) 1983 Eric P. Allman.  All rights reserved.
# Copyright (c) 1988, 1993
#       The Regents of the University of California.  All rights reserved.
#
# By using this file, you agree to the terms and conditions set
# forth in the LICENSE file which can be found at the top level of
# the sendmail distribution.
#
#


#
#  This the prototype for a "null client" -- that is, a client that
#  does nothing except forward all mail to a mail hub.  IT IS NOT
#  USABLE AS IS!!!
#
#  To use this, you MUST use the nullclient feature with the name of
#  the mail hub as its argument.  You MUST also define an ‘OSTYPE’ to
#  define the location of the queue directories and the like.
#  In addition, you MAY select the nocanonify feature.  This causes
#  addresses to be sent unqualified via the SMTP connection; normally
#  they are qualified with the masquerade name, which defaults to the
#  name of the hub machine.
#  Other than these, it should never contain any other lines.
#

divert(0)dnl
```

```
OSTYPE(linux)
FEATURE(nullclient, fred.$m)
Cm example.com
```

이제 당신은 다음의 명령을 이용하여 sendmail.cf 파일을 생성해야 한다.:

```
make example.master.cf example.slave.cf
```

그리고 이 파일들을 각각의 고유한 컴퓨터마다 sendmail.cf 라는 이름으로 복사해 넣어야 한다.

대부분의 sendmail의 설정 파일은 /etc/sendmail 밑에서 설정이 놓여진다. sendmail에 대하여 virtusertable.db와 genericstable.db의 2가지 다른 파일을 사용하는 것은 이 설정의 이유이다. 이와 같은 특별한 파일들을 이용하여, 그들의 모 파일을 생성할 수 있다. 첫번째로 virtusertable.src 를 살펴 보자.:

```
John.Public@example.com                jpublic
Jane.Doe@example.com                   jdoe@somemachine.somedomain
abuse@example.com                      root
Pointyhaired.Boss@example.com          #phb#@hotmail.com
```

이것은 새로운 목적지로 보낸 메일 주소의 목록이다. John.Public@example.com 으로 메일을 보내는 것은 jpublic이라는 리눅스의 계정으로 넘겨 주는 것을 의미한다. Jane.Doe@example.com으로 보내어진 메일은 다른 메일 계정인 jdoe@somemachine.somedomain로 넘겨지고 이것은 외부 서버로도 메일을 되보내는 것이 가능함을 의미한다. abuse@example.com로 보내어진 메일들은 같은 방식으로 root에게 보내어진다. 다른 파일인 genericstable.src를 살펴보자.:

```
jpublic                                John.Public@example.com
janedoe                                Jane.Doe@example.com
whgiii                                 Pointyhaired.Boss@example.com
```

이 파일은 네트웍에서 생성된 메일을 외부로 나갈 수 있는 메일로 이름을 바꾸는 역할을한다. 이것은 메일 발신을 위해 메일 주소를 내부에서 통용되는 당신이 선택한 계정 이름 대신 jdoe@somemachine.somedomain 식으로 바꾸는 작업을 수행한다. 마지막으로, 다음의 Makefile을 /etc/sendmail/ 안에 생성하자.:

```
all : genericstable.db virtusertable.db

virtusertable.db : virtusertable.src
        makemap hash virtusertable < virtusertable.src

genericstable.db : genericstable.src
        makemap hash genericstable < genericstable.src
```

make를 기동시켜 sendmail을 사용할 수 있게 하는 해쉬 파일을 생성하라. 그리고 설정 파일을 바꾼 후에는 언제라도 다시 make 를 한 후 sendmail을 재시작해야 한다는 것을 기억하라.

7.3.2 다른 메일 프로그램을 이용한 방법

개인적으로는 sendmail만을 사용하고 있다. 이 섹션을 자신이 쓰기를 원하는 누구라도 좋으니 연락을 주기 바란다. 다시 말해서 나는 MTAs 와 같은 Postfix, Exim, 혹은 smail 등에 대한 더 자세한 것은 나중에 말하겠다는 것이다. 물론 나는 그런 프로그램들을 사용해 본 다른 누군가가 이 섹션을 맡아 써 주는 것이 낫다고 생각한다.

7.4 홈페이지 공간 설정

당신은 당신의 보안을 위해 게이트 웨이의 뒤쪽으로 사설 네트웍의 외부에서 보이는 웹 서버를 셋업하게 된다. 만약 웹 서버가 사설 네트웍 안의 다른 컴퓨터의리소스로서 사용되는 데이터베이스에 접근하는 것이 필요 하다면, 이 상황은 네트웍에서 부터의 보안상 좀 더 복잡한 상황이다. 그런 상황의 설정은 이 문서의 범위를 벗어나는 것이다.

서버 설정의 세부적인 내용은 리눅스 WWW HOWTO의 아파치 문서를 찾아 보는 것이 좋다. 그 문서는 *metalab.unc.edu/pub/Linux/docs/HOWTO/WWW-HOWTO* `<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/WWW-HOWTO>` 에서 찾을 수 있다. (그러나 우리 나라에서는 KLDP에서 찾아 보면 되겠죠?)

7.5 FTP 설정

다시 한 번 말해 두지만, 당신의 FTP 서버는 외부로 노출된 컴퓨터여야만 하지, 사설 네트웍 게이트웨이 하의 컴퓨터여서는 안 될 것이다. 당신의 FTP 데몬 패키지의 수송에 따라 셋업의 방향을 따르도록 하다. 오래된 버전에서의 알려진 보안상의 약점에 대비하기 위해 최신 버전을 다운로드 받아 설치하도록 하라. 만약 당신의 FTP가 익명 사용자의 파일 업로드를 지원하지 않을 것이라면 데몬의 일부 옵션을 꺼야 한다는 것을 주지하자. 나는 승인된 사용자의 FTP 로그인이 FTP 호스트에 허락되지 않게 하고 당신이 필요로 하는 사용자만을 scp, 어떤 파일이 업데이트 되었을 때 그들이 FTP 호스트에 접속해 있어야만 하는 보안 쉘 원격 복사 명령을 사용할 것을 추천한다. 이것은 사용자에 대한 보안 설정에 대한 것이고 기타의 것이나 문제점들에 관한 것은 "당신의 도메인에 대한 보안"을 참고하라.

7.6 패킷 필터링의 설정

"방화벽 설정"부분에 자세한 내용이 있다.

제 8 절 당신의 도메인에 대한 보안

이 섹션은 당신의 새로운 도메인을 위한 보안의 설정을 말할 것이다. 강조될 부분은 바로 유저들에게 있다. 만약 당신의 보안이 너무나 강요되고, 인터페이스가 유저에게 너무 어렵게만 되어 있다면 유저들은 온전한 도메인에 타협하여 자신의 환경을 개발할 것이다. 그런 사태를 피할 가장 최적화된 방법은 가능한한 보안에 있어 투명성을 지향하라는 것이다. 그리고 유저들이 당신의 네트웍에 들어 와서 사이트의 보안 문제로 어려움을 겪을 때 용기를 북돋아주어라. 소위 유연성이라는 것이 중요한 것이다. 나는 보안이 너무나 엄격하여, 유저들이 그들의 외부로 나가기 위한 방화벽을 통한 네트웍 터널을 단순하게 설정할 수 밖에 없을 때를 알고 있다. 이것은 원격 로그인을 허용하는 것 보다는 좋다. 혹은 유저들이 그렇게 하도록 할 수도 있다. 어느쪽을 선택하건 당신의 자유다.

이 섹션은 당신의 네트웍이 외부로부터 공격을 당하거나, 혹은 내부의 스푸핑을 당하는 경우로 보안을 분류한다. 내부의 합법적인 사용자의 공격을 막는 것이 더욱 어렵고 고된 작업을 포함한다. 그리고 구체적인 내용은 이 문서를 넘어서는 것이다.

이 섹션에서 말하려는 보안의문제는 "적의 있는 라우터"에 대응하기 위한 것이다. 당신의 ISP의 라우터 공급자는 그것을 원격으로 설정, 제어할 수 있는 것일 가능성이 크므로, 관리자의 패스워드를 공급자에게서 얻어 설정한다. 예전에는 라우터의 생산자가 내장한 패스워드(관리자가 패스워드를 잊어버렸을 때를 대비한 응급용.) 가 시스템 크래커에게 알려져 보안의 문제가 생겼었다. 가능할 때 당신은 당신의 보안에 있어 라우터가 어떤 적의 있는 공격을 받음을 가정하여 디자인해야 한다. 그것은, 당신의 공식적인, 혹은 사설 네트웍의 어느 IP를 이용하여 다른 사이트에 패킷을 보냄으로서 누가 그 일을 했는지를 알 수 없게 하는 것이 있다.

8.1 방화벽 설정

이 섹션은 ipchains 기반의 매스커레이딩 설정과, 포워딩, 라우터의 필터링으로 나눈다. 당신은 IPCHAINS-HOWTO 를 먼저 읽는 것이 좋다. 그때 이것을 힌트의 *metalab.unc.edu/pub/Linux/docs/HOWTO/IPCHAINS-HOWTO* `<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/IPCHAINS-HOWTO>` 추가를 위해 읽도록 하자. 그 HOWTO는 매스커레이딩을 지원하는 커널의 컴파일부터, 이진 ipchains 사용의 세부 사항까지 다루고 있다. 당신은 외부 IP를 가진 어떤 컴퓨터도 방화벽으로서 할 수있다.

사설 네트워 게이트웨이 머신을 가정하고 당신의 스타트업 스크립트를 체크하라.:

1. 외부 이더넷 카드가 인식된다
2. ipchains에 따라 방화벽이 작동된다.
3. 포워딩이 작동된다.
4. 네트웍 서비스 데몬이 기동된다.

자, 예를 들어 슬랙웨어 기반의 시스템에서, 방화벽 설정은 rc.inet1과 rc.inet2 사이에서 이루어진다. 드물게 어떤 문제가 방화벽 설정 작업 중 일어난다면, 경고 메세지가 뜰 것이니 네트웍 서비스 데몬을 기동하기 전에 외부로 나가는 이더넷 회선을 빼도록 하라.

ipchains 기반의 방화벽에서 일반적인 문제는 당신의 룰대로 도착하는 루프백 인터페이스에서의 패킷들의 정정에 대한 권태 혹은 외부 혹은 내부에서의 방화벽 도착에 관한 것이다. 이런 지역적인 패킷은 방화벽에 의해 블록화된다. 아주 자주, 이런 작업은 방화벽의 어플리케이션이 방화벽의 호스트에서 돌고 있는 동안 집어내는 설정에 의해 설정산탄총 디버깅식 접근에 의한 소트에 의해 수리된다. 불운하게도 이런 것들의 방화벽에서의 결과는 의미없어지는 구멍이 된다. ipchains와 함께 이런 것은 방화벽의 스크립트를 디버깅이 용이하도록 쓰고 많은 패킷 소스 문제를 살펴보는 것으로 해결할 수 있다. 이것은 /sbin/firewall.sh 스크립트의 샘플이다.:

```
#! /bin/sh
#
# New firewalling script using IP chains. Creates a filtering router
# with network masquerading.
#

# define a few variables

IPCHAINS=/sbin/ipchains

LOCALNET="192.168.1.0/24"   # the private network
ETHINSIDE="192.168.1.1"             # fred.example.com's private IP #
ETHOUTSIDE="10.1.1.9"               # fred.example.com's public IP #
LOOPBACK="127.0.0.1/8"
ANYWHERE="0/0"
OUTSIDEIF=eth1                  # fred.example.com's private interface

FORWARD_PROCENTRY=/proc/sys/net/ipv4/ip_forward

#
# These two commands will return error codes if the rules
# already exist (which happens if you run the firewall
```

```
# script more than once). We put the commands before "set -e"
# so that the script doesn't abort in that case.

$IPCHAINS -N outside
$IPCHAINS -N portmap

set -e                  # Abort immediately on error setting
                        # up the rules.


#
# Turn off forwarding and clear the tables

echo "0" > ${FORWARD_PROCENTRY}

$IPCHAINS -F forward
$IPCHAINS -F input
$IPCHAINS -F output
$IPCHAINS -F outside
$IPCHAINS -F portmap


#
# Masquerade packets from within our local network destined for the
# outside world. Don't masquerade packets which are local to local

$IPCHAINS -A forward -s $LOCALNET -d $LOCALNET -j ACCEPT
$IPCHAINS -A forward -s $ETHOUTSIDE -d $ANYWHERE -j ACCEPT
$IPCHAINS -A forward -s $LOCALNET -d $ANYWHERE -j MASQ

#
# Set the priority flags. Minimum delay connections for www, telnet,
# ftp, and ssh (outgoing packets only).

$IPCHAINS -A output -p tcp -d $ANYWHERE www -t 0x01 0x10
$IPCHAINS -A output -p tcp -d $ANYWHERE telnet -t 0x01 0x10
$IPCHAINS -A output -p tcp -d $ANYWHERE ftp -t 0x01 0x10
$IPCHAINS -A output -p tcp -d $ANYWHERE ssh -t 0x01 0x10


#
# Anything from our local class C is to be accepted, as are
# packets from the loopback and fred's external IP.
$IPCHAINS -A input -s $LOCALNET -j ACCEPT
$IPCHAINS -A input -s $LOOPBACK -j ACCEPT
$IPCHAINS -A input -s $ETHOUTSIDE -j ACCEPT



# We'll create a set of rules for packets coming from the big, bad
```

```
# outside world, and then bind all external interfaces to it. This
# rule will be called "outside"
#
# We also create a "portmap" chain. The sockets used by daemons
# registered with the RPC portmapper are not fixed, and so it is
# a bit difficult to set up filter rules for them. The portmap
# chain is configured in a separate script.


#
# Send packets from any outside interface to the "outside"
# rules chain. This includes the $OUTSIDEIF interface and any
# ppp interfaces we create for dialout (or dialin).

$IPCHAINS -A input -i ${OUTSIDEIF} -j outside
$IPCHAINS -A input -i ppp+ -j outside


##################################################
#
#  Set up the "outside" rules chain              #
#
##################################################

#
# Nobody from the outside should claim to be coming from our localnet
# or loopback

$IPCHAINS -A outside -s $LOCALNET -j DENY
$IPCHAINS -A outside -s $LOOPBACK -j DENY

#
# No packets routed to our local net should come in from outside
# because the outside isn't supposed to know about our private
#  IP numbers.

$IPCHAINS -A outside -d $LOCALNET -j DENY

#
# Block incoming connections on the X port. Block 6000 to 6010.

$IPCHAINS -l -A outside -p TCP -s $ANYWHERE -d $ANYWHERE 6000:6010 -j DENY

#
# Block NFS ports 111 and 2049

$IPCHAINS -l -A outside -p TCP -s $ANYWHERE -d $ANYWHERE 111 -j DENY
$IPCHAINS -l -A outside -p TCP -s $ANYWHERE -d $ANYWHERE 2049 -j DENY
$IPCHAINS -l -A outside -p UDP -s $ANYWHERE -d $ANYWHERE 111 -j DENY
$IPCHAINS -l -A outside -p UDP -s $ANYWHERE -d $ANYWHERE 2049 -j DENY
```

```
#
# Block XDM packets from outside, port 177 UDP

$IPCHAINS -l -A outside -p UDP -s $ANYWHERE -d $ANYWHERE 177 -j DENY


#
# Block the YP/NIS port 653
$IPCHAINS -l -A outside -p TCP -s $ANYWHERE -d $ANYWHERE 653 -j DENY

#
# Don't bother logging accesses on TCP port 80, the www port.

$IPCHAINS -A outside -p TCP -s $ANYWHERE -d $ANYWHERE 80 -j DENY

#
# Accept FTP data and control connections.

$IPCHAINS -A outside -p TCP -s $ANYWHERE 20:21 -d $ANYWHERE 1024: -j ACCEPT

#
# Accept ssh packets

$IPCHAINS -A outside -p TCP -s $ANYWHERE -d $ANYWHERE ssh -j ACCEPT

#
# Accept DNS packets from outside

$IPCHAINS -A outside -p TCP -s $ANYWHERE -d $ANYWHERE 53 -j ACCEPT
$IPCHAINS -A outside -p UDP -s $ANYWHERE -d $ANYWHERE 53 -j ACCEPT

#
# Accept SMTP from the world

$IPCHAINS -A outside -p TCP -s $ANYWHERE -d $ANYWHERE 25 -j ACCEPT

#
# Accept NTP packets

$IPCHAINS -A outside -p UDP -s $ANYWHERE -d $ANYWHERE 123 -j ACCEPT

#
# Accept no tap ident packets, we don't use them

$IPCHAINS -A outside -p TCP -s $ANYWHERE -d $ANYWHERE 113 -j DENY

#
# Turn off and log all other packets incoming, TCP or UDP, on privileged ports
```

```
$IPCHAINS -l -A outside -p TCP -s $ANYWHERE -d $ANYWHERE :1023 -y -j DENY
$IPCHAINS -l -A outside -p UDP -s $ANYWHERE -d $ANYWHERE :1023 -j DENY

#
# Check against the portmapper ruleset

$IPCHAINS -A outside -j portmap


###############################################
#
#    End of "outside" rules chain             #
#
###############################################


#
# Block outgoing rwho packets

$IPCHAINS -A output -p UDP -i $OUTSIDEIF -s $ANYWHERE 513 -d $ANYWHERE -j DENY

#
# Prevent netbios packets from leaving

$IPCHAINS -A output -p UDP -i $OUTSIDEIF -s $ANYWHERE 137 -d $ANYWHERE -j DENY
#
# Turn on forwarding

echo "1" > ${FORWARD_PROCENTRY}
```

방화벽은 외부에서 들어오는 패킷만을 상대하는 것이 아니라 당신의 내부 네트웍에서부터 나가는 rwho나 netbios 의 네트웍 정보를 담은 패킷들도 감시한다.

미리 말해 두었어야 하는데 포트매퍼의 규칙은 약간 다른다. 그것은 포트매퍼 자신이 기록된 포트매퍼 데몬 기록이 어떤 포트가 열려 있는지를 말하기 때문이다. 특수한 데몬에 의해 사용되는 포트는 당신의 RPC 서비스 사용이 바뀜에 따라 바뀔 수 있거나 혹은 그들의 기동 명령 전환에 따라 바뀔 수 있다. 이와 같은 내용의 스크립트인 /sbin/firewall.portmap.sh 는 포트매퍼 데몬을 위한 규칙을 따르고 있다.

```
#! /bin/sh
#
ANYWHERE=0/0

IPCHAINS=/sbin/ipchains

$IPCHAINS -F portmap

# Rules for preventing access to portmapped services by people on the outside
#
/usr/bin/rpcinfo -p | tail +2 | \
        { while read program vers proto port remainder
          do
```

```
                prot=`echo $proto | tr "a-z" "A-Z"`
                $IPCHAINS -l -A portmap -p $prot -s $ANYWHERE -d $ANYWHERE $port -j DENY || exit
          done
        }
```

우리는 내부 네트웍에서 적법한 패킷이 날아오는 것을 걱정할 필요가 없으며 포트맵 체인은 오직 외부에서 날아오는 것만을 확인한다.

방화벽 설정 로그는 kern.info와 함께 하는 klogd를 통하여 의심스러운 것들을 기록에 남긴다. 이것은 일반적인 접속 시도에도 마치 스텔스처럼 몰래 살핀다.

자, 우리는 이런 것들을 다 알게 되었다.우리는 시스템이 시작되는 동안 작은 윈도의 약점이 없음을 확신하는 것이 좋다. 그러므로 우리는 우리의 시작 과정을 다음과 같이 설정할 필요가 있다.:

```
#! /bin/sh
#
# Get the network started, securely
#
#
/etc/rc.d/rc.inet1              # Configure the network interfaces
                                # and set up routing.
/sbin/firewall.sh || { echo "Firewall configuration failed"
                       /sbin/ifconfig eth1 down }

/sbin/ipchains -I outside 1 -j DENY     # Deny all incoming packets

/etc/rc.d/rc.inet2              # Start the network daemons

sleep 5                         # Let them stabilize

# Secure the portmapped services
/sbin/firewall.portmap.sh || { echo "Portmap firewall configuration failed"
                               /sbin/ifconfig eth1 down }

/sbin/ipchains -D outside 1       # Allow incoming packets
```

eth1 이 외부로 보여지는 IP를 할당받고 있다고 가정하자. 어떤 ipchains 규칙이 설정에 실패하였다면, 경고 메시지가 뜰 것이다. "외부의" 체인은 방화벽의 서비스는 포트매퍼의 서비스가 기동되기 전에는 사용할 수 없는 규칙이므로 네트웍 서비스 데몬이 기동되기 전에는 모든 패킷을 거부할 것이다. 포트맵 서비스가 방화벽의 역할을 하며, 외부의 체인을 재인식하는 것이다.

8.2 SSH1 설정

이 글을 쓰는 시점에서 OpenSSH는 내가 여기 언급하는 특징 중 하나를 제공하지 않는다. 그러나 OpenSSH는 아주 활발히 발전되고 있으므로, 이것은 언제라도 바뀔 수 있는 부분이다. 빠져 있는 까다로운 특징은 당신이 scp, ssh, slogin 등을 rcp, rsh, rlogin 등의 이름으로 바꾸어, rcp, rsh, rlogin 등의 원래의 프로그램이 ssh의 클라이언트 프로그램으로 바뀌어, sshd의 작동 없이는 사용할 수 없게 하는 설정에 관한 것이다. rsh를 사용하며 기원을 하는 대신 ssh 클라이언트 프로그램을 사용하여 사용자들의 보안 문제를 간단히 해결할 수 있는 것이다. 모든 이들의 스크립트로, rdist 설정과 원격의 sshd가 작동하는 원격 사이트에서 수정없이 작업을 지속할 수 있다. 그러나 데이터는 암호화되어 보내지며 이것이 강력한 확인이 되는 것이다.

웹 사이트*www.ssh.org/* <http://www.ssh.org/>에서 ssh1을 구하고, 그것을 컴파일하여 암호화되지 않은 r-프로그램들(rsh, rlogin, rcp 등)과 대체하라. 먼저, 그 세 가지 파일들을 /usr/lib/rsh에 복사해 넣고, ssh 패키지를 다음과 같이 설정하라.:

```
./configure --with-rsh=/usr/lib/rsh/rsh --program-transform-name='s/^s/r/' --prefix=/usr
```

설명에 따라 실행 파일을 설치하고 설정한다. 사설 네트웍 게이트웨이에서 sshd 설정은 다음과 같은 엔트리를 정의할 것이다.:

```
ListenAddress 192.168.1.1        # fred's internal IP
IgnoreRhosts no
X11Forwarding yes
X11DisplayOffset 10
RhostsAuthentication no
RhostsRSAAuthentication yes
RSAAuthentication yes
PasswordAuthentication yes
```

당신은 /etc/sshd_config 파일 안에 다른 엔트리를 설정해야만 하게 될 것이다. 그러나 그 필드를 바꾸지 않도록 하자. 당신이 이 파일 안에 당신이 고려할 모든 엔트리를 갖고 있다면, 그 엔트리 파일을 새 파일인 /etc/sshd_config.ext 에 외부의 네트웍을 위해 복사해 넣어라. 새 파일에서 다음 두 개의 필드는 수정하여라. :"ListenAddress"는 사설 네트웍의 IP를 외부로 보여질 수 있는 이름으로 교체하는 것이다. 예를들면 10.1.1.9가 fred.example.com으로 바꾸는 것이 있겠다. 그리고 "PasswordAuthentication"은 "no"로 설정해라. 당신의 네트웍을 기동하는 스크립트에서, sshd를 2번 시작하도록 하라. 한 번은

```
/usr/sbin/sshd
```

와 같이, 다시 한 번은

```
/usr/sbin/sshd -f /etc/sshd_config.ext
```

의 식이다.

이것은 2개의 sshd 데몬을 기동시킨다. 하나는 내부 인터페이스의 로그인 시의 패스워드를 체크하지만 다른 것은 외부 인터페이스에서 RSA키를 누군가가 로그인 하기 전에 포함하게 한다.

다음으로, 내부로 들어오는 telnet과 셸 서비스를 inetd 설정 파일에서 끄도록 한다. 이 부분은 방화벽 설정에 관한 섹션에서 이비 외부에서의 접근을 말할 때 언급한 바 있다. 그러나 이것은 방어에 있어서만 좋은 생각이다. 모든 작업을 순조로이 할 수는 없을 지도 모른다는 뜻이다.

집에서, 혹은 도시 밖에서 로그인을 원하는 사람들은 RSA 키가 필요하다. 그들은 어떻게 해야 할 지 알고 있으며, 그들은 telnetd를 당신의 방화벽상의 평범한 포트에 두는 것과 같은 다른 방법으로 그런 일을 하여 자신의 에너지를 소모하고 싶어하지 않는다.

RSA 키의 생성은 다음과 같은 명령으로 이루어진다.:

```
ssh-keygen -b 1024 -f new_rsa_key
```

당신은 패스 페이즈로부터 힌트를 받을 것이다. 이것은 공백이 되어서는 안 될 것이다. 파일 new_rsa_key로 접근하고, 패스 페이즈를 알고 있는 어떤 사람이 모든 RSA 인증 과정을 통과하기 위한 모든 필요 조건을 갖고 있는 것이다. 패스 페이즈는 유추해 낼 수 없는 패스워드이거나 일반적이지 않은 긴 문장 이어야 한다. 파일 new_rsa_key는 플로피 디스크, 랩탑애 복사될 수 있으며 패스 페이즈에 속하여 계정에 로그인하는 사람에 대해 특정한 RSA 키를 허가한다.

특수한 RSA키에 의해 계정에 대한 접근 허가가 이루어지는 설정에서, 가장 단순한 생성은 사설 네트웍 게이트웨이(이 컴퓨터는 로그인 접근을 돌려 보낼 것이다.)에 사용자를 위한 $HOME/.ssh/ 디렉토리를 생성하여 $home/.ssh/authorized_keys 파일 안의 ssh-keygen 명령으로 인하여 생성된 new_rsa_key.pub 를 복사해 넣는 것이다. sshd 매뉴얼 페이지에서 신뢰하는 IP, 혹은 호스트 이름을 가진 곳에서 오는 로그인 요청이나 혹은 외부에서 보내어진, 인증으로 통해 신뢰가 가능한 명령의 경우에서 당신이 추가해 넣을 수 있는 다른 옵션 키들에 관한 설명으로서 나와 있는 “AUTHORIZED_KEYS FILE FORMAT” 섹션을 보면 잘 나와 있다. (예를 들자면, RSA 키의 다른 곳으로, 혹은 누군가에게 메일로 보내어 백업하는 경우 등.)

RSA 키 메카니즘을 가능한한 사용자 우선으로 만드는 데 이제 한 가지가 남았다. 만약 어떤 사용자가 한두 번의 과정을 거쳐 패스 페이즈를 알아 내게 된다면, 그는 스스로가 보안의 구멍이 되어 보안상의 문제를 일으킬 가능성을 스스로의 손 안에 갖고 있는 것이다. 리눅스에서, 로그인 쉘은 ssh-agent 기반으로 불러진다. 예를 들어 만약 업무용으로 사용되는 회사의 랩탑에서 실수로 xdm이 실행되어 사용자에게 X 세션의 권한이 넘어가게 된다면, /var/X11R6/lib/xdm/Xsession_0 을 열어 시동될 때 불러지는 다음의 행을 가능하면 다음과 같이 바꾸도록 하라.:

```
exec "$startup"
```

이 행을 이렇게 바꿔라.:

```
exec ssh-agent "$startup"
```

나의 xdm 설정에서, 그 파일에서 각각의 3줄이 바뀌어졌다. 사용자가 랩탑에 로그인할 때, 그는 다음의 명령을 입력해야만 하는 것이다.

```
ssh-add new_rsa_key
```

로그인 프롬프트가 떠서 패스 패이즈를 입력 받아서부터, 사용자가 그의 X 세션을 랩탑에서 종료할 때 까지, 사설 네트웍 안의 모든 윈도에서 패스 페이즈 없이 접근할 수 있기 위해서 해야 할 일이다.

sshd 가 당신의 사설 네트웍상의 모든 컴퓨터에서 외부의 호스트에 대하여 작동하고 있다. 사설 네트웍 상의 것이 아닌 컴퓨터에 대하여 ListenAddress 엔트리를 /etc/sshd_config 안에 “0.0.0.0”와 같이 설정할 수도 있다. 당신은 다음 명령을 통해 호스트 키를 설정할 수 있다:

```
ssh-keygen -b 1024 -f /etc/ssh_host_key -N ""
```

make-ssh-known-hosts를 실행하고 /etc/ssh_known_hosts 파일 안에 사설 및 검증된 네트웍의 모든 컴퓨터를 분류하는 것이다.

암호화되지 않은 r-서비스들과 텔넷의 외부에서 안으로 들어오는 접속을 무력하게 한다. 텔넷의 실행 파일을 지울 것은 없는 것이 이것은 포트 23번 상의 다른 텔넷 세션들보다 유용한 것이니까. 당신은 사설 네트웍 상에서 패스워드를 이용한 인증을 사용하여 외부로부터의 접속을 차단하고 외부의 호스트에서 보내어진 RSA키를 로그에 남길 수도 있다.

이것은 사설 네트웍상의 호스트드이 각각의 /etc/hosts.equiv 파일 상에 기록되어 있을 경우 사용자들에게 편리한 것이다. sshd 데몬은 그것들에 의해 사람들의 rlogin과 rsh를 컴퓨터들 간에 패스워드나 패스 페이즈 없이 가능하게 한다. 모든 접속에 있어서, 컴퓨터들은 호스트 레벨 RSA 키를 통하여 각각의 동일성을 증명해야만 하는 것이다.

사용자가 외부 네트웍의 IP를 가진 컴퓨터에서부터 사설 네트웍 상의 컴퓨터로 로그인을 하기를 원할 때 다른 것이 나타난다. 당신은 /etc/hosts.equiv 혹은 $HOME/.shosts를 패스워드 인증 과정 없이 이용할 수 없게 된다. 그것은 사용자가 검증되지 않은 IP-이것은 매스커레이딩 된 것일수도, 방화벽일 수도 있지만 호스트 키가 일치하지는 않을 것이다. -를 가진 곳에서 들어 왔기 때문이다. 이에는 두 가지 해법이 있다. 한 가지는 당신이 /etc/hosts.equiv 나 $HOME/.shosts 메소드를 사용할 것을 주장할 경우인데, 이 때는 사용자들이 사설 네트웍에 로그를 남겨야만 할 것이다. 그리고 이 로그는 들어오기를 시도한 외부의 컴퓨터에도 남을 것이다. 다른 방법은 RSA 키 안증을 이용하는 것이다. 그것은 어떤 IP에서 호스트 이름으로 lookup을 시도하는 부주의에도 언제나 작동한다.

8.3 X의 설정

많은 유저들이 보안보다는 편리함을 추구하기 때문에 계속 여러 모의 탐색을 해 나가고 있다. 이것은 많은 사람들이 다음과 같이 실행하게 한다.

```
xhost +
```

이 명령은 X를 초기화 하는 것이다. 이런 허가를 받은 X서버는 세계의 누구라도 접근할 수 있게 된다. 외부의 어떤 임의의 사용자가 당신의 루트의 윈도 회면을 당신이 지정한 것에서 당신의 상관이 자기 어머니에게 사무실을 구경시켜 주다가 경악하게 할 만한 것으로 바꾸어 놓을 수도 있게 된다는 뜻이다. 이런 외부인은 당신의 모니터를 제어하고 당신의 스크린 상에 띄워지는 내용을 넘겨 볼 수도 있는 것이다. 쓸 데 없는 잔소리지만, 이것은 당신이 다른 사이트의 로그인로서 패스워드를 넘기는 것 혹은 민감한 사안의 문서를 화면에 띄워 수정하는 상황을 생각하면 좋을 게 없는 상황이라는것을 쉽게 알 수 있을것이다. xhost 프로토콜 자신은 본래부터 사용자 기반으로 화면을 사용할 권한을 양도하는 것이 불가능한 한계를 갖고 있고, 오직 기계 기반인 것이다.

xauth 인증에 들어가자. 만약 당신이 xdm 을 가지고 있다면 당신은 아마 이미 xauth 인증을 실행하고 있을 것이디만 xhost가 여전히 돌고 있다. 그리고 아마도 사람들은 컴퓨터 사이에서 X의 프로세스를 사용하고 있을 것이다. 다시 말하자면, 이것의 결론은 사용자들이 xhost 명령을 더 이상 사용하지 않고도 사용하기 쉽게, 보안과 편리함을 함께 누리자는 것이다.

"SSH1 설정" 섹션에서 "X11 포워딩" 을 기본 지식으로 하여 sshd 셋업을 묘사한 것은 xhost 테크닉보다 사용하기 쉬운 것이다. 당신이 당신의 터미널에 접속할 때 당신은 간단히 rlogin으로 원격지 컴퓨터에 들어가 넷스케이프나 xv, 혹은 다른 좋아하는 것들을 $DISPLAY 변수를 조정하거나 접근 권한을 얻지 않고도 사용할 수 있다. ssh 로그인은 사용자에게 투명한 방식으로 설정하고, 각각의 당신의 X 패킷에 관한 암호화는 그들이 네트웍을 떠나기 전까지 지속된다.

만약 당신이 sshd X11포워딩을 어떤 이유로 이용할 수 없다면, 당신은 xauth를 당신이 당신의 X 서버에 접근하기를 제한하는 다른 컴퓨터들에 대한 인증책으로 쓸 수 있다. 사용자를 위한 혹은 특별한 그들을 도울 수 있는 셀 스크립트들이 기술되어 있는 문서들이 있다. "jpublic" 컴퓨터상에서는 "barney"의 당신의 X 서버에 접근하기 위한 관련된 명령으로 다음이 있다.:

```
/usr/X11/bin/xauth extract - $DISPLAY | rsh -l jpublic barney /usr/X11/bin/xauth merge -
```

나는 xhost를 당신의 컴퓨터 엔트리에서 지우려는 유혹을 받는 편이다. 만약 그것이 어떤 프로그램에 문제가 된다면 당신은 최소한 그것이 보안에 약한 것이라는 점은 알 수 있는 것이다. 이것은 xauth 시퀀스 리스트를 사용하는 xhost를 위한 되돌려 놓는 셸 스크립트를 작성하는 것으로 충분하다.

rsh이 ssh 프로그램을 암호화 하지 않았을 때, xauth 키는 단순한 텍스트로서 보내질 뿐이라는 점을 기억하자. 그것을 입수한 누구라도 당신의 서버에 접근할 수 있다. 그러므로 당신은 암호화 과정을 위해 ssh를 사용하지 않았다면 더 많은 보안을 기대해서는 안 되는 것이다. 게다가 사용자의 홈 디렉토리가 NFS 로 외부에 노출되어 있다면, xauth 키는 그 누구라도 NFS 패킷을 통해 채어 갈 수 있다는 점을 기억하고 ssh를 당신의 시스템에서 기동시켜야 한다는 것을 생각하자.

8.4 디스크 공유 설정

서버로 메일이 왔을 때 그것을 어떤 곳에서도 읽고 메일을 보낼 수 있게 한다면 편리할 것이다. 그러나 심심하고 따분한 나머지 못된 짓을 해 보려 서성대는 일반 유저들에 대한 약간의 주의는 기울여야 한다. AUTH_DES 의 실행 없이 NFS를 사용하는 것은 무방비 상태나 다름없다. NFS의 클라이언트 에 대한 신뢰관계는 접근을 보장하는 것이다. 그것은 서버에서의 패스워드 인증 없이도 클라이언트에서 각각의 개인인 자신의 파일에 접근할 수 있다는 것이다. 윈도의 경우에는 유닉스 식의 파일의 접근 제한을 완벽하게 무시하고 NFS 적인 공유를 어떤 uid 없이도 가능하게 한다. 따라서 NFS는 리눅스 박스 나 유닉스처럼 당신의 즉각 조정이 가능한 하에서만 설정해야 한다. 물론 윈도로 듀얼 부팅이 되는 컴퓨터에도 해선 안 될 것이

다. 만일 당신이 메일 스풀 디렉토리나 혹은 어떤 다른 디렉토리를 때때로 윈도 박스로 이용되기도 하는 컴퓨터와 공유하기를 원한다면 그때는 “security=USER” 모드에 의해 보안이 입증되는 삼바(samba)를 이용하기 바란다. 당신의 네트웍에 허브로 컴퓨터를 연결하는 것 보다는 스위치 라우터를 이용하는 것도 약간의 장난과 악의로 윈도가 깔려 있는 컴퓨터를 사용하는 사람들을 대비하는 데 도움이 될 것이다. 어쨌건 네트워크 상으로 공유되는 어떤 디스크의 보안을 유지하는 것은 아주 어려운 일이라는 것만 명심하라.

그런데도 정말로 네트웍에 연결된 디스크의 보안을 철저히 하고 싶은가? 대부분 확실한 방어법은 이슈가 된다. 만약 당신이 기밀이 적힌 서류를 책상 위에 두고 나갔을 때 누군가가 사무실에 들어가 그 기밀을 보았다고 하자. 그는 즉시 그것이 어느 정도의 가치가 있는것인지를 생각하고는 그것이 정말 기치 있는 것이라면 인간 본성의 어두운 부분에 따라 책상에 앉아 그것을 읽을 것이다. 만약 그 서류가 파일 캐비닛이나 책상 서랍 속에 들어 있었다면 그것은 분명히 더 어려운 일이었을 것이다. 어떤 단순한 네트웍에서의 보안의 목적은 누구도 우연히 그 보안을 깨게 하지 않는 것에 있다 해도 과언이 아니다.

제 9 절 정보(번역해야 할 필요성을 못 느낀 부분. 실은 귀찮아서. ^^* 죄송.)

This document was written as internal documentation for the DYNACAN project, as part of the project's continuing development under the control of the Ministry of Human Resources Development Canada.

This document has benefited considerably from the suggestions of

? Rod Smith (*rodsmith@rodsbooks.com* <mailto:rodsmith@rodsbooks.com>), who suggested I provide details on registering a domain name and on setting up with a dynamic IP, and pointed me at the various dynamic IP hosting services and at Granite Canyon.

? Greg Leblanc (*gleblanc@my-deja.com* <mailto:gleblanc@my-deja.com>) for useful suggestions on improving the clarity of the document.

? Sami Yousif (*syousif@iname.com* <mailto:syousif@iname.com>), who told me about www.dhs.org.

? Marc-Andr?Dumas (*m_a_dumas@hotmail.com* <mailto:m_a_dumas@hotmail.com>), who suggested the section on moving your domain to a new IP number.

제 10 절 용어 해설

다음은 이 문서에 언급되었던 몇몇 단어와 약자들에 대한 설명이다.

CGI 스크립트

CGI는 A Common Gateway Interface Script의 약자이다. 이것은 웹 페이지의 내용을 생성하는 명령이 들어 있는 프로그램이다. 만일 단순한 내용과 그래픽이 보여 지는 웹페이지가 있다면, 당신은 정렬을 하거나 문서를 좀 더 다이나믹하게 만들어 주는 CGI 스크립트의 사용을 원할 것이다. 이런 것의 예를 들면 게시판이나 피드백 폼, 혹은 쇼핑 카트(장바구니 기능) 같은 것이 있을 것이다.

DHCP

Dynamic Host Configuration Protocol의 약자. RFC 1531에 정의되어 있는 IP 번호와 넷마스크, 게이트 웨이 등을 중앙 서버에서 처리하여 이루어 지는 컴퓨터 간의 TCP/IP 네트웍 상의 표준을 따른다. 관리자로서 이와 같은 컴퓨터의 설정을 하는 것이 네트웍을 추가할 때 서버에서의 쉬운 처리 면에서 더 낫다.

DNS

Domain Name Service의 약자. IP 번호를 도메인 이름으로 전환하는 것의 표준이 된다. 중앙 데이터베이스를 이용한다.

DSL

Digital Subscriber Line의 약자. 고속 네트웍 연결 관계에서 전화 접속을 일반적인 것으로 넘기는 역할을 한다.

유동 IP

주기적이며 서로 다른 IP 번호를 기반으로 하고 있다. 한번 사용한 번호가 계속 유지된다는 보장이 전혀 없다. 유동 IP는 당신이 네트웍에 연결하기 위해 전화를 걸어 재접속 할 때 마다 다른 번호가 부여된다. 혹은 DHCP의 양도에 의해 주기적으로 바뀌어 나갈 수도 있다. 일반적으로 telnet 이나 ssh 등의 일반적인 서비스들은 당시의 접속이 이루어 졌을 당시의 IP가 바뀌었을 때 서비스되지 않는다.

DNS 쿼리 포워딩

IP번호를 도메인 네임으로 전환하는 "DNS" 상의 질의

FTP

파일 전송 프로토콜. 인터넷 상에서 한 컴퓨터에서 다른 컴퓨터로 파일을 보내는 가장 기본적인 방법.

ftpd

호스트에서 "FTP" 서비스를 공급하는 데몬. 원격의 클라이언트의 요청을 받아들인다.

Internet Service Provider는 "ISP" 를 보라.

IP 는 "IP 번호" 부분을 보라.

IP 번호

확정된 네트웍의 "할당된 주소" 를 말한다. ipv4라고 불리우는 주소 지정의 표준을 따르는 이 숫자는 4개의 8비트 값으로 이루어져 일반적으로 사용되는 십진수로 기록된다. 인터넷에 연결된 컴퓨터가 IP 번호를 통해 정보가 담긴 패킷들을 보내게 된다.

ISP

인터넷 서비스 공급자. 당신의 네트웍 접속 전반을 공급하는 회사를 일컫는 말이다. (하드웨어, 서비스 호스팅, IP 번호의 제어 등을 포함하는 전반.)

매스커레이딩

하나의 컴퓨터에서 다른 컴퓨터로 보내어지는 패킷을 필터링 하는 그 헤더에 중심이 되는 컴퓨터의 정보를 기록하는 것. 중심이 되는 컴퓨터는 원래 보낸 컴퓨터로 그 회신을 돌려 보낸다. 하나의 IP번호를 네트웍 상의 컴퓨터가 공유하여 매스커레이딩 호스트를 통하여 외부로 나갈 수 있다.

named

도메인 네임 서버. 이것은 "DNS" 쿼리에 답변하고 BIND 패키지의 부분을 배분한다.

Network Time Protocol "NTP" 편을 보라.

NTP

Network Time Protocol. 당신의 컴퓨터 시계를 실제의 표준시로 정정한다. 시계의 "정확도가 높은" 시계들을 기준으로 하고 있다.

OS 오퍼레이팅 시스템의 약자. 운영 체제를 말한다. 리눅스나 윈도, BeOS , BeOS, HP-UX 등등.....

PHB

뾸난 머리의 상관 *www.unitedmedia.com/comics/dilbert/about/html/boss.html* `<http://www.unitedmedia.com/comics/dilbert/about/html/boss.html>`. 딜버트로 유명한 스코드 애덤즈의 창조물이다.

Provider

"ISP" 편을 보라.

DNS 쿼리의 역전

"DNS" 가 도메인 이름에서 그 IP를 추출해 내는 것을 말한다.

Router

IP 주소에 기반하여 패킷을 보내고 당신의 이더넷 하드웨어와 어떤 연결에서도 당신의 ISP와의 사이에 매개가 되는 특수한 하드웨어 디바이스를 말한다.

ssh

The secure shell의 약자. rlogin, telnet, ftp, 혹은 다른 서비스를 위한 강력한 암호화 기법. 스푸핑 공격 등 외부의 공격이나 혹은 패킷 스니핑을 차단한다.

고정되어 있는 IP

"IP 주소" 란 영구적으로 임대되거나 혹은 고정되어 있는 것이다. 당신이 그 주소에 대한 계약을 백지화하지 않는 이상 그 주소는 언제 라도 당신이 이용할 수 있는 것이고, 인터넷 상의 어떤 다른 컴퓨터도 같은 번호를 사용할 수는 없는 것이다. 그렇지 않은 경우를 "유동 IP"라고 말한다.