Transparent Proxy with Squid mini-HOWTO

Daniel Kiracofe v1.1, 2000년 9월 29일

황 성준sj@kldp.org 2001년 1월 3일

이 문서는 단지 리눅스와 squid만을 사용하여 트랜스페어런트 캐쉬(transparent cache)를 사용하는 HTTP 프록시 서버(HTTP proxy server)를 설정하는 방법에 관한 정보를 제공한다.

목 차

제 1 절 소개

1.1 코멘트

이 미니 하우투(mini HOWTO)에 관한 코멘트와 일반적인 피드백(feedback)을 환영하고, drk@unxsoft.com로 저자인 Daniel Kiracofe한테 메일을 띄울 수 있다.

1.2 저작권과 상호

- 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. (이 문서의 작성에 쓰인 SGML과는 별도로) 이 문서 내의 어떠한 소스 코드라도 GNU General Public License를 따르고, GNU archive로부터 익명의 FTP를 통해 구할 수 있다.

1.3 #include <disclaimer.h>

No warranty, expressed or implied, etc, etc, etc...

제 2 절 트랜스페어런트 프록시(Transparent Proxying)의 개요

2.1 동기 부여

”보통의” 프록시를 사용함에 있어서, 클라이언트는 웹 브라우징 소프트웨어(web browsing software)에서 프록시의 호스트명과 포트 번호를 지정한다. 그러면 브라우저는 프록시에 요청하고, 프록시는 원래의 서버한테로 요청을 전달한다. 이것은 모두 성능이 뛰어나지만, 때때로 여러 상황들 중의 하나가 생기도 한다. 가령,

- 네트워크 상의 클라이언트가, 원하든 원치 않든 간에, 프록시를 사용하도록 강요하기를 원한다.
- 클라이언트가 프록시를 사용하기를 원하지만, 그들이 프록시를 사용하는지 알리고 싶지 않다.
- 클라이언트가 프록시를 사용하고 있지만, 수 천 혹은 수 만의 웹 브라우저에서 설정을 변경하는 작업을 일체 원하지 않는다.

이런 상황에 트랜스페어런트 프록시 사용이 편리하다. 웹 요청은 프록시에 의해 가로채여질 수 있다. 즉, 클라이언트 소프트웨어가 알고 있기로는, 원래의 서버와 교신하겠지만, 실제로는 프록시 서버인 것이다.

시스코 라우터는 트랜스페어런트 프록시를 사용하는 것을 지원한다. 그러나, (상당히 놀랍게도) 리눅스는 라우터로 동작할 수 있고, TCP 연결을 지역 포트(local ports)로 재지정함으로써 트랜스페어런트 프록시 역할을 수행하기도 한다. 그렇지만, 우리의 웹 프록시가 재지정의 영향을 알도록 할 필요가 역시 있고, 그래서 적절한 원래의 서버한테 연결할 수 있다. 이런 작동에는 두 가지 일반적인 방식이 있다:

첫 번째는 여러분의 웹 프록시가 트랜스페어런트 프록시를 알지 못하는 때이다. 여러분은 웹 프록시의 전반부에 놓이고 여러분을 대신해서 온갖 귀찮은 사소한 일을 돌보는 트랜스프록시 (transproxy)라고 불리는 재빠르고 작은 데몬을 사용할 수 있다. 트랜스프록시는 John Saunders에 의해 작성되었고, ftp://ftp.nlc.net .au/pub/linux/www/ 또는 metalab 미러 사이트로부터 구할 수 있다. 트랜스프록시는 이 문서에서 더 이상 논의하지 않을 것이다.

더욱 완전한 솔루션은 트랜스페어런트 프록시 자체를 알고 있는 웹 프록시를 갖는 것이다. 여기서 우리의 관심을 모으는 것은 squid이다. squid는 유닉스 시스템을 위한 오픈 소스 (Open Source) 캐싱 프록시 서버이다. www.squid-cache.org로부터 구할 수 있다.

2.2 이 문서의 범위

이 문서는 squid 2.3 버전과 리눅스 커널 2.2 버전, 이 글이 쓰여지는 현재(2000년 3월)에 가장 안정된 공개본에 초점을 맞출 것이다. 또한 2.0과 더 최근의 2.1 리눅스 커널들만큼 일찍부터 squid와 같이 작동한다. 여러분이 더 이전의 발표본에 대한 정보가 필요하다면 www.unxsoft.com에서 좀 이전의 문서를 찾을 수 잇다.

만일 여러분이 리눅스 2.3을 사용하기 원한다면, ipchains 대신에 netfilter라고 불리는 것을 사용하여야 한다. 그렇지만, 여러분이 개발 커널을 실행하고 있다면, 제공된 문서로부터 여러분 자신의 방식으로 netfilter를 이해할 수 있다고 가정한다. 만일 그렇지 않다면, 여러분은 실제로 개발 커널을 실행해서는 안 된다. 리눅스 2.4가 발표되면 이 문서는 netfilter를 다루기 위해 갱신될 것이다.

이 문서는 HTTP 프록시를 사용하는 것만을 중점적으로 다룬다는 것을 알린다. 트랜스페어런트 FTP 프록시를 사용하는 것에 대한 묻는 전자우편을 수많이 받는다. 투명하게 FTP를 대리(프록시)하는 것이 이론적으로 불가능하지는 않지만, 반면에 HTTP보다 훨씬 더 어렵고, 그런 것이 가능한 어떤 도구에 대해 현재 구하는 방법을 알지 못한다. 만일 여러분이 잘 알고 있으면, 여러분도 자신의 HOWTO를 작성하도록 제안한다...

제 3 절 커널을 구성하기

먼저, 모든 적절한 옵션이 여러분의 커널에 제대로 설정되었는지 확인할 필요가 있다. 만일 여러분이 선택한 배포본으로부터 표준적인 커널을 사용하고 있다면, 트랜스페어런트 프록시 사용은 가능할 수도 있고 가능하지 않을 수도 있다. (IIRC, 그것은 RH 6.1에 있다. 그러나 그것에 관해서 나를 인용하지 말라.) 만일 여러분이 확신하지 못한다면, 말하기에 가장 좋은 방법은 이 섹션을 간단히 건너뛰는 것이다. 그리고, 만일 다음 섹션에서 나오는 명령어들이 여러분한테 별난 오류를 제공한다면, 그것은 아마 커널이 적절하게 구성되지 않은 때문일 것이다.

만일 여러분의 커널이 트랜스페어런트 프록시를 사용하도록 구성되지 않았으면, 다시 컴파일할 필요가 있을 것이다. 커널을 다시 컴파일 하는 것은 (적어도 맨 처음에는) 복잡한 과정이고, 이 문서의 범위를 넘어서는 것이다. 만일 커널을 컴파일 하여야 할 필요가 있다면, *The Kernel HOWTO* `<http://metalab.unc.edu/pub/Linux/docs/HOWTO/Kernel-HOWTO>`를 보라.

여러분의 구성에서 설정할 필요가 있는 옵션은 다음과 같다. (주의: 이것들 중 어떤 것도 모듈로서 만들어져서는 안 된다.)

- Sysctl support
- TCP/IP networking
- IP: firewalling
- IP: always defragment
- IP: transparent proxy support
- /proc filesystem support

일단 새 커널을 올려서 실행하고 있다면, IP forwarding을 가능케 할 필요가 있을 것이다. IP forwarding은 여러분의 컴퓨터가 라우터로 동작하는 것을 허용한다. 이것은 일반적인 수준의 사용자가 하기를 원하는 것이 아니기 때문에 기본적으로 꺼져 있고 명시적으로 실행 시에 가능케 해야만 한다. 그렇지만, 여러분의 배포본은 이미 여러분을 위해 이것을 할 수도 있다. 점검하기 위하여, "cat /proc/sys/net/ipv4/ip_forward"를 하라. 만일 "1"을 보면 잘 된 것이다.그렇지 않으면, "cat '1' > /proc/sys/net/ipv4/ip_forward"를 하라. 그 다음에 여러분은 /etc/rc.d/에 적당한 시동(bootup) 스크립트에 그 명령어를 추가하기를 원할 것이다.

제 4 절 squid를 설정하기

우리는 squid를 준비하고 실행시킬 필요가 있다. www.squid-cache.org로부터 최신의 소스 타볼(tarball)을 다운로드 한다. 개발 버전이 아닌, 안정적인(STABLE) 버전을 얻었는지 확인하라. 이 글이 쓰여지는 현재 최신의 것은 squid-2.3.STABLE4.tar.gz였다.

이제, 아카이브(archive)를 풀고(untar) 압축을 해제(gunzip) 하라 ("tar -xzf <filename>"를 사용하라) 자동구성(autoconfiguration) 스크립트를 실행하고("./configure"), 컴파일("make") 하고, 그 후에 설치하라("make install").

이제, (기본적인 설정을 바꾸지 않았다면, /usr/local/squid/etc/squid.conf 에 설치된) 기본적인 squid.conf 파일을 편집할 필요가 있다. squid.conf 파일은 많은 주석이 달려 있다. 사실상, squid를 위한 가장 좋은 문서 중의 하나는 squid.conf 파일 내에 있다. 여러분이 모든 것을 준비하고 실행한 후에 되돌아 가서 모든 것을 다시 읽어야 한다. 하지만, 우선은 최소 사항을 요청해 보자. 다음 지시자를 찾아서 주석을 없애고(uncomment), 적절한 값으로 변경하라:

- httpd_accel_host virtual
- httpd_accel_port 80
- httpd_accel_with_proxy on httpd_accel_uses_host_header on

끝으로, http_access 지시자를 보라. 기본적으로는 대개 "http_access deny all"이다. 이것은 어느 누구도 squid에 접근하는 것을 방지할 것이다. 우선은, 여러분이 "http_access allow all"로 변경할 수 있겠지만, 일단 작동하고 나면 아마 ACL (Access Control List)에 관해 읽고 여러분의 로컬 네트워크에 있는 사람들한테만 캐쉬(cache)에 접근할 수 있도록 캐쉬를 설정할 필요가 있을 것이다. 이것은 우습게 보일지 모르지만, 여러분의 캐쉬에 접근하는 데 있어 어떤 종류의 제한을 반드시 두어야 한다. (포르노 필터 또는 발언이 매우 자유롭지 못한 국가에서의 필터 등과 같은 선별(filtering) 방화벽 뒤의 사람들은 종종 광대한 개방된 프록시를 "강탈하고" 여러분의 대역폭을 몽땅 차지한다 .

"squid -z"로 캐쉬 디렉토리를 초기화 하라. (만일 이것이 squid를 새로 설치하는 것이 아니면 이 단계를 건너뛰어야 한다.)

이제, /usr/local/squid/bin/ 디렉토리에 있는 RunCache 스크립트를 사용하여 squid를 실행하라. 만일 작동한다면, 웹 브라우저의 프록시 설정을 해당 박스의 IP와 (기본적인 포트번호를 변경하지 않았다면) 포트 3128에 맞추고, 정상적인 프록시로서 squid를 접근할 수 있어야 한다.

구성하는 것을 돕는 추가적인 도움이 필요하다면 www.squid-cache.org에서 FAQ를 보라.

제 5 절 ipchains 설정하기

ipchains는 거의 모든 최근의 배포본 (커널 2.2에 기반한 것)과 함께 설치되어진다. 그렇지만, 여러분이 ipchains를 가지지 않았다면 ftp://ftp.rustcorp.com/ipchains/로부터 구할 수 있다. ipchains는 매우 강력한 도구이고, 여기서는 수박 겉핥기에 불과할 것이다. 더 많은 정보를 원하면 ipchains HOWTO에 관한 http://www.rustcorp.com/linux/ipchains/HOWTO.html를 보라.

규칙을 설정하기 위하여 두 가지를 알아야 할 필요가 있다. 박스의 IP 주소(하나의 예로서 나는 192.168.1.1을 사용할 것이다)와 squid가 실행 중인 포트가 그것들이다.

첫 번째로, 우리는 이 박스의 실제적인 웹서버로 향하는 패킷을 모두 허용할 필요가 있다. 우리는 루프백(loopback) 인터페이스와 이더넷(ethernet) 인터페이스를 둘 다 설정하여야 한다. 여러분의 박스에 실제적인 웹서버가 없다고 하더라도 이 단계를 건너뛰어서는 안 된다. 이런 규칙의 부재는 프록시가 그 자신한테 연결을 시도하는 무한한 전달(forwarding) 반복을 만들 수 있기 때문이다. 다음과 같은 명령어를 사용하라:

- ipchains -A input -p TCP -d 127.0.0.1/32 www -j ACCEPT
- ipchains -A input -p TCP -d 192.168.1.1/32 www -j ACCEPT

이제, 트랜스페어런트 프록시를 사용하기 위한 주문(magic words)이다:

- ipchains -A input -p TCP -d any/0 www -j REDIRECT 3128

여러분은 /etc/rc.d/ 아래의 적절한 시동 스크립트에 위의 명령어들을 추가할 필요가 있을 것이다.

제 6 절 모두 한꺼번에 하기

만일 모든 것이 지금까지 제대로 되었다면, 다른 장비로 가서 게이트웨이를 새로운 squid 박스의 IP로 바꾸고 인터넷을 사용해 보라. 그런 요청이 원래 서버에 직접 전해지는 대신에 실제로는 여러분의 프록시를 통해 전달되는 것을 확인하려면, /usr/local/squid/logs/access.log 로그 파일을 점검하라.

제 7 절 후속 자원들

여러분한테 아직 도움이 필요하다면, www.squid-cache.org에서 squid FAQ 또는 squid 메일링리스트를 확인하라. 또한, drk@unxsoft.com로 나한테 전자우편을 보낼 수도 있다. 시간이 허용하면 여러분의 질문에 답하도록 노력할 것이다. (때때로 하기도 하고, 때때로 그렇게 하지 않을 것이다) 제발, 제발, 제발, "ipchains -L"의 결과와 설정 파일의 관련된 부분을 메일로 보내 달라. 혹은, 그렇지 않을 경우에, 아마도 여러분을 많이 도와 주지 못하게 될 것이다...