djbdns DNS

임은재

eunjea (at) kldp.org

v0.1, 2001/12/23

이 문서는 BIND (http://www.isc.org/products/BIND/)를 대체하는 djbdns (http://cr.yp.to/djbdns.html)에 관한 문서입니다. 이 문서의 마지막 버전은 http://kldp.org/~eunjea (http://kldp.org/~eunjea/)에 있습니다.

고친 과정

고침 0.12001-12-22고친이 ej

첫번째 문서.

1. 들어가기

1.1. DNS가 어떻게 작동하는가?

어떻게 내 컴퓨터가 예를 들어 network-surveys.cr.yp.to의 IP 주소를 찾아내는가?

이 IP를 찾기위해 당신의 컴퓨터는 인터넷상의 일련의 **DNS** 서버들과 통신하게 되는데, 예를 들어 중앙 **root** 서버중 하나인 192.5.5.241는 다음과 같은 정보를 가지고 있다.

```
.:192.5.5.241
&to:198.6.1.82
```

두번째 줄이 의미하는 것은 **delegation** (위임)이다. 즉, ".to 에 관한 것은 198.6.1.82에 물어보세요"라는 뜻이다. 198.6.1.82 주소를 가지는 DNS 서버는 다음과 같은 정보를 가지고 있다.

```
.to:198.6.1.82
&yp.to:131.193.178.181
```

역시 두번째 줄에서 .yp.to 에 관한것은 131.193.178.181 주소가 가지고 있다는 것을 알수 있다. 131.193.178.181 주소의 DNS 서버는 다음과 같은 정보를 가진다.

```
.yp.to:131.193.178.181
=network-surveys.cr.yp.to:131.193.178.100
```

두번째 줄에서 network-surveys.cr.yp.to의 IP 주소를 얻었다.

당신의 컴퓨터는 파일로 저장되어 있는 루트 서버의 주소를 시작으로 위의 예와 같이 차례대로 서버들에 접속한 끝에 ip 주소를 가져오게 된다. 당신은 이런 과정이 반복되는 것을 피하기 위해 **DNS cache**를 운영할수 있다.

하드웨어적 문제나 네트워크의 문제가 야기될 경우를 대비해서 여러개의 root서버와 .to 서버가 존재한다. 각 루트 서버는 다음과 같은 정보를 가지고 있다.

```
.:198.41.0.4:a
.:128.9.0.107:b
.:192.33.4.12:c
.:128.8.10.90:d
.:192.203.230.10:e
.:192.5.5.241:f
.:192.112.36.4:g
.:128.63.2.53:h
.:192.36.148.17:i
.:198.41.0.10:j
.:193.0.14.129:k
.:198.32.64.12:l
.:202.12.27.33:m
&to:128.250.1.21:a
&to:193.0.0.193:b
&to:196.7.0.139:c
&to:206.184.59.10:d
&to:198.6.1.82:e
&to:206.86.247.253:f
&to:148.59.19.11:g
```

.to 서버는 다음과 같은 정보를 가진다.

```
.to:128.250.1.21:a
.to:193.0.0.193:b
.to:196.7.0.139:c
.to:206.184.59.10:d
.to:198.6.1.82:e
.to:206.86.247.253:f
.to:148.59.19.11:g
&yp.to:131.193.178.181:a
# or, in BIND master zone-file format:
# yp.to IN NS a.ns.yp.to
# a.ns.yp.to IN A 131.193.178.181
```

루트 서버는 무작위로 선택되며 이 서버로 부터 다시 무작위로 .to 서버로 이동하게 된다.

1.2. 리버스(reverse) DNS는 어떻게 작동하는가?

어떻게 IP 주소 208.33.217.122로 부터 호스트 이름을 찾을까?

당신의 컴퓨터는 일련의 DNS 서버에게 122.217.33.208.in-addr.arpa에 대해 질의를 하게 된다. 루트 서버는 다음과 같은 정보를 가지고 있다.

```
&33.208.in-addr.arpa:206.228.179.10:c
&33.208.in-addr.arpa:144.228.254.10:b
&33.208.in-addr.arpa:144.228.255.10:a
```

144.228.254.10 DNS 서버는 다음 정보를 가지고 있다.

```
.33.208.in-addr.arpa:144.228.255.10:a
.33.208.in-addr.arpa:206.228.179.10:c
.33.208.in-addr.arpa:144.228.254.10:b
&217.33.208.in-addr.arpa:209.191.164.20:a
&217.33.208.in-addr.arpa:206.253.194.65:b
```

209.191.164.20 DNS 서버는 다음 정보를 가지고 있다.

```
.217.33.208.in-addr.arpa:209.191.164.20:a
.217.33.208.in-addr.arpa:206.253.194.65:b
=mm-outgoing.amazon.com:208.33.217.122
```

2. djbdns DNS tools

djbdns (http://cr.yp.to/djbdns.html)는 qmail (http://cr.yp.to/qmail.html)의 제작자인 D. J. Bernstein이 만든 Domain Name System 도구이다.

djbdns는 보편적으로 사용되는 BIND (http://www.isc.org/products/BIND/)(Berkeley Internet Name Domain)를 완전히 대체할수 있으며, 더 안전하고 관리하기 손쉽다. 또한 각각의 역할에 따라 서로 다른 작은 프로그램들로 나누어져 있으므로 요구에 따라 필요한 것만 사용할수 있는 장점을 가지고 있다.

djbdns는 제작자가 보안 구멍에 상금 (http://cr.yp.to/djbdns/blurb/security.html)을 걸어 놓았을 만큼 안전을 보증하고 있다. (일반적으로 널리 쓰이지 않고 많이 알려지지 않은 것이 더 안전하다는 것은 ftpd,SMTP 서버등에도 적용된다).

나는 여기서 어떤 DNS 서버가 더 낳다거나 하는 등의 말은 하지 않겠다. 어떤 DNS 서버를 사용할 것인지는 완전히 당신의 판단에 달려있다. 이와 관련된 플레임 워가 있었는데 관심있는 사람들은 읽어보길. http://www.isc.org/ml-archives/bind-users/2000/08/msg01075.html

이 문서에서는 DNS 자체에 대한 설명(문서 참고 [1])은 하지 않으며, djbdns 가 포함하고 있는 모든 프로그램과 기능들을 설명하지 않으므로, 더 자세한 것은 각자 문서를 참고하거나 이 문서의 새버전이 나올때까지 기다린다. ;-)

다음과 같은 상황을 예로 djbdns의 dnscache와 tinydns의 설치와 설정을 설명한다.

```
                linux box                    192.168.1.10
   Internet    +-----------+                |--------------- C1
```

```
---------------|-----       |              | 192.168.1.20
   1.2.3.4     |            | 192.168.1.1  |--------------- C2
               |      ------|--------------| 192.168.1.30
               +-----------+               |--------------- C3
```

위는 요즘 자주 볼수 있는 설정으로 리눅스 박스에는 두개의 인터페이스로 하나는 인터넷과 연결되어 있으며, 나머지는 내부망에 연결되어 있다.

이 리눅스 박스는 NAT (문서 참고[2])로 내부망의 클라이언트들에게 인터넷 연결을 공유하고 또한, 도메인을 가지고 있어 웹,메일,ftp 서버의 역할도 하고 있다고 가정한다. 이때 인터넷 공인 ip와 내부망에 연결된 ip는 각각 1.2.3.4 와 192.168.1.1 라고 가정한다.

3. 필요한 것

- djbdns (http://cr.yp.to/djbdns/djbdns-1.05.tar.gz)
- daemontools (http://cr.yp.to/daemontools/daemontools-0.76.tar.gz)
- 시간
- 커피 한잔

4. 유저 추가하기

설치될 프로그램들이 안전하게 운영되도록 다음 유저를 추가해 준다. 이 유저의 권한으로 각 데몬들이 실행되게 할것이다.

```
useradd -M dns
useradd -M dnslog
```

5. daemontools 설치

daemontools는 각종 서비스들을 관리하기 위한 도구들의 모음으로 ucspi-tcp (http://cr.yp.to/ucspi-tcp.html)와 같이 사용하면 inetd 나 xinetd 같은 수퍼 데몬을 완전히 대체할 수 있다. 더 자세한 것은 각각의 웹 페이지를 참고하고, 다음과 같은 순서로 설치한다:

```
mkdir -p /package
chmod 1755 /package
cd /package

tar xzf daemontools-0.76.tar.gz
rm daemontools-0.76.tar.gz
```

```
cd admin/daemontools-0.76

# 마지막으로 install을 실행해주면 컴파일과 설치가 진행된다.
package/install
```

설치가 끝나면 /service 와 /command 디렉토리가 추가로 만들어진 것을 볼수 있으며, /etc/inittab 에는 재부팅시 데몬툴이 시동되도록 다음과 같은 라인이 추가된 것을 볼수 있다:

```
SV:123456:respawn:/command/svscanboot
```

설치가 끝나면 svscanboot가 자동으로 시작된다 (BSD 시스템은 재부팅). 프로그램의 제작자인 D. J. Bernstein 에게 리포트도 하나 보내자.

```
mail djb-sysdeps@cr.yp.to < /package/admin/daemontools/compile/sysdeps
```

6. DNS Cache 설치

외부 네임 서버를 사용하지 않고 직접 dns cache를 운영할때의 장점은 내부망(또는 리눅스 박스 자체)이 요구하는 도메인 네임 resolving을 처리하고 이것을 캐쉬에 저장했다가 다음번 질의에 빠르게 응답할수 있기 때문이다.

dnscache는 BIND 와는 달리 cache 전용 데몬으로 유연한 설정과 더 빠른 응답 속도를 보인다. 특히 도메인을 가지고 있지 않으며 인터넷 공유의 용도로만 사용된다면 이것만 설치하면 된다 (이 경우 pdnsd[3]를 사용할수도 있다).

6.1. 서비스 디렉토리

dnscache 서비스 디렉토리는 dnscache-conf 를 사용하여 다음과 같이 만든다. 192.168.1.1 은 내부망에 연결된 ip 주소이다.

```
dnscache-conf dns dnslog /var/dnscache 192.168.1.1
```

6.2. 서비스 시작

svscan 으로 하여금 새로 만든 서비스를 시작하도록 /service에 링크를 걸어준다.

```
ln -s /var/dnscache /service
```

이제 5초 이내에 dnscache 데몬이 시작될 것이다.

6.3. 설정

이제 해당 네트워크로 부터의 질의에 응답하도록 파일을 하나 만들어 준다. 우리는 192.168.1.* 의 질의를 허용하도록 할것이므로:

```
touch /var/dnscache/root/ip/192.168.1
```

만약 접속을 허용할 네크워크 대역이나 IP가 더 있다면 같은 방법으로 설정해 주면된다.

6.4. resolv.conf 수정

마지막으로 리눅스 박스 자체가 dnscache를 사용하도록 /etc/resolv.conf 를 다음과 같이 수정한다.

```
nameserver 192.168.1.1
```

이제부터 192.168.1.x로 부터 들어오는 모든 도메인 resolving 쿼리에 응답하고 캐쉉하게 되므로, 클라이언트들이 192.168.1.1을 네임서버로 사용하도록 설정을 수정한다. 로그는 /var/dnscache/log/main/current 에 남는다.

6.5. 캐쉬 메모리 조절

리눅스 박스 뒤쪽에 클라이언트가 매우 많고, 네임 resoving 처리를 매우 많이 하는 경우 dnscache가 사용할 cache 메모리를 더 늘려줄수 있는데 만약 10 Mbyte 정도를 사용하도록 할것이라면 다음과 같이 한다.

```
echo 10000000 > /service/dnscache/env/CACHESIZE
echo 10485760 > /service/dnscache/env/DATALIMIT
```

dnscache가 이미 실행중이라면 재시동한다.

```
svc -t /service/dnscache
```

6.6. 외부 DNS 캐쉬 사용하기

DNS Cache가 방화벽뒤에 위치하고 있으며 특정 DNS 서버로의 쿼리외의 UDP 패킷은 drop되도록 설정되어 있거나, 로컬 DNS Cache의 쿼리가 느려서 외부 DNS 서버를 사용하도록 하려면?

1.03 버전이후의 부터는 다음과 같이 할 수 있다.

```
echo 1 > /service/dnscache/env/FORWARDONLY
```

이제 루트 서버들의 IP가 적혀 있는 "/service/dnscache/root/servers/@"의 내용에 외부 DNS서버의 IP만을 적어준다. 이제 부터 dnscache는 resolve 요청이 있을때 답을 자신이 가지고 있지 않을때는 여기에 적어준 외부 DNS 서버에 쿼리하고 그 결과를 캐쉬, 응답하게 된다.

dnscache가 실행중이 였다면 재시작 해주어야 한다.

```
svc -t /service/dnscache
```

6.7. BIND와 dnscache를 같이 운영하기

다음장에서와 같이 dnscache는 DNS 서버와 동일한 서버에서 운영할 수 있으므로, 역시 BIND와도 같이 운영할 수 있다. 다음과 같이 BIND가 특정 IP에만 바인딩되도록 /etc/named.conf를 설정해 주면 된다. 예를 들어 IP 주소가 1.2.3.4라면:

```
options {
interface-interval 0;
listen-on { 1.2.3.4; };
};
```

BIND를 재시작한다.

7. DNS server

우리의 리눅스 박스가 도메인 이름을 가지고 있으며, 네임 서버를 운영해야 한다면 tinydns를 사용한다.

7.1. 서비스 디렉토리

tinydns-conf를 사용하여 서비스 디렉토리를 다음과 같이 만든다.

```
tinydns-conf dns dnslog /var/dns 1.2.3.4
```

7.2. 서비스 시작

svscan 으로 하여금 새로 만든 서비스를 시작하도록 /service에 링크를 걸어준다.

```
ln -s /var/dns /service
```

5초 이내에 tinydns 데몬이 시작될 것이다.

7.3. 설정하기

리눅스 박스의 도메인이 kldp.org 라 한다면 다음과 같이 설정해준다.

```
cd /var/dns/root/
./add-ns kldp.org 1.2.3.4
./add-ns 3.2.1.in-addr.arpa 1.2.3.4
```

이제 웹/메일/ftp 서버등도 돌아가고 있으므로, 다음과 같이 설정한다. 예:

```
cd /var/dns/root/

./add-host www.kldp.org 1.2.3.4
./add-alias kldp.org 1.2.3.4
./add-alias ftp.kldp.org 1.2.3.4
./add-mx mail.kldp.org 1.2.3.4

make
```

이 설정들은 /var/dns/root/data 에 텍스트 형태로 저장되며 수정후에는 tinydns가 사용할수 있는 cdb 형태로 만들어 주기 위해 make 나 tinydns-data 를 실행한다.

이미 설치한 dnscache가 참조하도록:

```
echo "1.2.3.4" > /var/dnscache/root/servers/kldp.org
```

7.4. 확인하기

이제 까지의 설정이 제대로 되었는지 tinydns-get으로 확인해보자. tinydns-get은 현재 디렉토리의 data.cdb 파일을 읽어 정보를 출력한다.

```
$ cd /service/dns/root/
$ tinydns-get a kldp.org

1 kldp.org:
77 bytes, 1+1+1+1 records, response, authoritative, noerror
query: 1 kldp.org
answer: kldp.org 86400 A 1.2.3.4
authority: kldp.org 259200 NS a.ns.kldp.org
additional: a.ns.kldp.org 259200 A 1.2.3.4
```

8. 저작권, 기타

8.1. 저작권 정보

8.2. 책임의 한계

본 저자는 문서의 내용이 야기할 수 있는 어떠한 결과에 대해서도 책임을 지지 않습니다.

8.3. 마치며

시간나는 대로 djbdns에 포함되어 있는 다른 툴들(walldns, rbldns등)에 대한 설명을 추가하도록 하겠습니다. 수정,추가,의견 환영합니다.

러시아, 모스크바에서 임은재.

8.4. 참고 문서

이 문서는 djbdns의 제작자인 D. J. Bernstein (http://cr.yp.to/djb.html)이 쓴 관련 문서(2001년 12월 23일 기준)를 참고로 작성되었습니다.

주석

1. http://kldp.org/KoreanDoc/html/PoweredByDNS-KLDP/
2. http://kldp.org/Translations/html/2.4_NAT-KLDP/2.4_NAT-KLDP.html
3. pdnsd (http://home.t-online.de/home/Moestl/index.html) 역시 caching 전용 서버로 캐쉬한 내용을 디스크에 저장해 놓거나 하는 등의 몇가지 재미있는 기능을 가지고 있다.