

# djbDNS Kurulumu NASIL

Yazan:  
**Ömer Faruk Şen**  
<ofsen (at) enderunix.org>

2 Ağustos 2001

## Özet

Bu belgede şu an Unix dünyasında bulunan en güvenli dns sunucusu olan djbDNS ve kurulumu anlatılacaktır.

Bu belgenin özgün sürümüne [http://www.enderunix.org/docs/djbdns\\_installation.html](http://www.enderunix.org/docs/djbdns_installation.html) adresinden erişebilirsiniz.

## Konu Başlıkları

<b>1. Giriş</b>	4
<b>2. DNS Nedir?</b>	4
2.1. Resolving (Çözümleme): Aranılan bir kaydı bulma işlemi	4
2.2. Authoritative Nameserving (Yetkili İsim Sunumu)	5
<b>3. Önceden Kurulması Gerekli Yazılımlar</b>	5
<b>4. Dnscache Programının Kurulması</b>	6
<b>5. Tiny-dns Kurulumu</b>	7
<b>6. /service/tinydns/root/data Dosyasının Biçemi</b>	9
<b>7. Bölge (zone) Transferi</b>	10
<b>8. axfrdns-get (Bölge Transfer İstemcisi)</b>	11
<b>9. Djbdns Komutları</b>	11
9.1. dnscache	12
9.2. tinydns	12
9.3. walldns	12
9.4. dnsq	12
9.5. dnsqr	12
9.6. dnsfilter	12
9.7. axfrdns	12
9.8. dnstrace	13
9.9. dnstracesort	13
9.10. axfr-get	13
<b>10. Yardımcı Bilgiler</b>	13
10.1. Aynı veriyi içeren 5 web sunucusu için yük dengelemesini nasıl yaparım?	13
10.2. Değişik istemcileri değişik sunuculara nasıl yönlendirebilirim?	13
10.3. Aynı ağda bulunun bir dnscache o ağ için tanımlı yetkili isim sunucusunu nasıl bulacak?	14

10.4. djbDNS çalışmadı...	14
<b>11. Kaynaklar</b>	14
<b>12. Yasal Açıklamalar</b>	15
12.1. Telif Hakkı ve Lisans	15
12.2. Feragatname	15

## Geçmiş

---

1.0	2 Ağustos 2001	ÖFŞ
-----	----------------	-----

---

## 1. Giriş

BIND, the Buggy Internet Name Daemon is like Microsoft Windows. The damn thing doesn't work. Every version has been brimming with bugs.

-- D.J.Bernstein (<http://cr.yp.to/djbdns/ad/unbind.html>)

Bu belgede şu an Unix dünyasında bulunan en güvenli dns sunucusu olan djbDNS ve kurulumu anlatılacaktır.

DjbDNS D.J.Bernstein tarafından yazılmıştır. Kendisi aynı zamanda qmail, tcpserver, publicfile, daemontools gibi çeşitli sistem yazılımları yazmıştır. Sanırım üstteki ifadeye katılmamak mümkün değil. DjbDNS gerek üniversite gerekse ticari şirketler tarafından rahatça kullanılabilir. Bind tek başına SPOI (Single Point of InSecuritness) lakabını yeterince hak etmektedir. Ayrıca djbdns dns sistemi için yeni istemci kitaplıklarını da sunmaktadır. <http://cr.yp.to/djbdns/ad/library.html> adresinden gerekli bilgiler edinebilir.

OpenBSD üzerine kurulacak basit bir makinada djbdns 2-3 sene kesintisiz hizmet verebilir. OpenBSD'yi merak ediyorsanız [http://www.enderunix.org/documents/openbsd\\_installation.html](http://www.enderunix.org/documents/openbsd_installation.html) adresinde TÜRKÇE kurulum kılavuzundan yararlanarak bu işletim sistemine bir giriş yapabilirsiniz.

Bilindiği üzere şu anda dünya üzerinde en çok kullanılan dns sunucu yazılımı olan Bind paketinde bir çok güvenlik açığı bulunmaktadır. Bunların ortaya çıkması sadece zaman meselesidir. Nitekim şu an kullanılan bind sürümünde bile birkaç açığın var olduğu ve bunun Internet'te istismar edildiği konuşulmaktadır. Söylenti bile olsa bind'i kullanmak zorunda olan sistem yöneticileri ne demek istediğini anlayacaklardır. Securityfocus.com adresinde de djbdns kurulumu anlatılmaktadır. Zaten djbdns ile biz böyle tanıştık. Sendmail'den Qmail'e çekilen zorlukları bind'tan djbdns'e geçerken yaşamayacaksınız. Bind ile djbdns'nin karşılaştırmasını <http://cr.yp.to/djbdns/ad/easeofuse.html> adresinde bulabilirsiniz.

## 2. DNS Nedir?

Domain Name System [DNS] (Alan Adı Sistemi) dağıtık yapıda bir veritabanıdır. Bu sistem makina isimlerini IPv4 (ya da ipv6) adreslere ya da IPv4 adresleri makina isimlerine çevirmeye yarar. Ayrıca bir alan için gerekli eposta sunucusunun adreslerini ya da alan adı sunucularının hangi IP adreslerinde bulunduğunu daha teknik bir ifade ile RR'ları (Resource Records – Özkaynak Kayıtları) tutar. DNS yapısı gereği hızlı olmalıdır. Bu yüzden yapılan sorgulamalar eğer 512 bayttan küçük ise ya da yapılan işlem asıl DNS sunucusu ile yardımcı DNS sunucusu arasında bölge transferi değilse UDP üzerinden yapılacaktır. DNS sisteminin yazarları DNS için gerekli bilgileri diğer Internet araçlarında olduğu gibi RFC'lerde tanımlamışlardır (RFC 1034 ve 1035). Bu belgelere bakarsanız içinde bir kaç kod ve bölge soayalarının nasıl yazılacağını bulursunuz.

BIND'de (Berkeley Internet Name Daemon) yazılırken bu kriterler dikkate alınmıştır. Fakat bu sistemde gereksiz karmaşıklıkta kodlar ve bunun getirdiği güvenlik problemleri vardır. Fakat djbdns bu karmaşadan uzak ve güvenli bir yazılımdır. Bunun sonucu olarakta "two-thumbs-up bir daemon" olmaya bizce hak kazanmıştır. Hatta bizce bind 9.1.X ve ileriki sürümlerinde kullanılan "Dns-sec sistemi" bile bind için bir çözüm olmaktan çok uzaktır. Sonuçta temel yani "daemon Internet" in ilk zamanlarından kalmıştır. Şu an IPv4'te yaşanan güvenlik sorunları da aynı sebepten oluşmaktadır.

Bu yüzden köklü bir değişim olması gerekliliği savunulmuş ve IPv6 ortaya atılmıştır. Bind içinde benzer model söz konusu olmadıkça djbdns popülaritesini sürdürecektir. (Haklı bir popülarite !!!).

Genel olarak dns yazılımlarının yaptığı iki vazife vardır. Bind kullananlar eğer çok profesyonel değillerse ya da bu konu ile alakalı kitap okunmamışsa bu iki olguyu karıştırabilmektedirler. Bunlar:

### 2.1. Resolving (Çözümleme): Aranılan bir kaydı bulma işlemi

Mesela `http://www.yildiz.edu.tr/` adresine karşılık gelen IPv4 adresinin `193.140.1.1` olmasının bulunması. Çözümleme yapan yazılımlar iki çeşit işlem yaparlar; ardışık çözümleme ve ardışık olmayan çözümleme. Sorgularda gönderilen RD (recursion required – ardışık gerekli) bitlerine göre sorgunun türü belirlenir. Ardışık olmayan sorgulara cevap veren sunucular cevap olarak ardışık isim sunucuları verirler.

Sonuç olarak yapılan bir sorgu ardışık ise `http://www.yildiz.edu.tr/` için doğrudan `193.140.1.1` IP'si ya da "makina bulunamadı" cevabı verilebilir. Fakat ardışık olmayan bir sorguda cevabı bulmak için başka bir isim sunucusunun IP'sini verebilir. Ardışık olmayan sorgular aynı zamanda yinelemeli sorgular olarak da bilinirler.

## 2.2. Authoritative Nameserving (Yetkili İsim Sunumu)

Bir alan hakkında bilgi bulunduran sunucudur. Mesela `yildiz.edu.tr` alanının MX, NS, A (Bunlar – Resource Record – Özkaynak Kaydı olarak bilinir) kayıtlarının tutulduğu isim sunucusudur.

### CANLI BİR ÖRNEK

Bir sorgu nasıl yapılır. Mesela `http://www.yildiz.edu.tr/` adresinin IPv4 karşılığı nedir? Bunun için çözümleyici ya da sizin bilgisayarınız Internet'te birkaç yetkili isim sunucusunu gezmek zorundadır. İlk başta ". " sunucularından `tr` TLD'si için yetkili isim sunucu IP'si bulunur. Bunun IP'si (ki bu IP'nin makina ismi ülkemiz için `ns1.metu.edu.tr` ya da `ns2.metu.edu.tr`'dir) alındıktan sonra bu yetkili isim sunucuya "edu.tr." i barındıran yetkili isim sunucu IP'si sorulacak ve bu işlem böylece devam edecektir ve sonuç olarak `193.140.1.1` adresini sizin makinanız ya da `/etc/resolv.conf` dosyanızda bulunan çözümleyicinin `yildiz.edu.tr`'nin yetkili isim sunucusundan aldığı bilgilere göre belleğinde belli bir müddet tutacaktır. Bu süreye TTL (time to live – yaşam süresi) denir. Bu süre boyunca çözümleyici gerekli bilgileri belleğinde saklayacaktır. Djbdns için aksi belirtilmezse bu süre 42 dakikadır. Kurulacak göre bu süre 5 saniye de yapılabilir. (Nitekim birden fazla `www` sunucunuz var ve bunlar arasında döner turnuva kullanarak yük dengelemesi yapmak istersek bu saniye değeri kullanılacaktır).

## 3. Önceden Kurulması Gerekli Yazılımlar

Djbdns'i kullanmak için önceden kurulması gereken birkaç yazılım vardır. Bunlar yine D.J. Bernstein tarafından yazılmışlardır. Bunlar kullanılması istenen sisteme göre değişir. Genel olarak sistem eğer sadece asıl sunucu olacaksa `daemontools` kurulması yeterli olacaktır. Daemontools sisteminizde çalışan servisleri gözetleyen bir yazılımdır. Fakat sadece gözetlemekle kalmaz eğer yapılandırma dizininde bulunan bir servis çalışmıyor ise bunları tekrar başlatır. Bu yazılım ile benzer vazifeleri yapan birkaç yazılım daha vardır. Mesela "mon" örnek verilebilir.

Kurulumu:

```
$ wget daemontools-0.70.tar.gz
$ tar xzvf daemontools-0.70.tar.gz
$ cd daemontools-0.70
# make setup check
# mkdir /service
# svscan /service &
```

Bundan sonra asıl yazılımımız olan djbdns sisteme kurulur.

```
$ wget djbdns-1.05.tar.gz
$ tar xzvf djbdns-1.05.tar.gz
$ cd djbdns-1.05
# make setup check
```

Eğer kurulacak olan sistemde ikinci bir sunucu; yani bir yardımcı sunucu olacaksa bölge transferi gerekecektir. Bunun için gerekli olan yazılımlar `Ucspi` (tcpserver ) ve `axfrdns` olacaktır. Bu yazılımlarda üstte anlatıldığı şekilde kurulabilir.

## 4. Dnscache Programının Kurulması

`Dnscache` yukarıda anlatılan yazılımlardan çözümleyici kısmına girmektedir. Yani `/etc/resolv.conf` dosyasına yazılacak IP şimdi kuracağımız yazılımın çalıştığı bir makineye ait olmalıdır. Burada kuracağımız `dnscache` aslında dnscache'in kurulması olası iki çeşidinden birisi olacaktır. Bunlar "harici dnscache" ve "dahili dnscache" olarak adlandırılır. Biz burada Internet'de ya da ağıımızda bulunan başka bir makinenin sorgularını işleme yetisine sahip olan "harici dnscache" kuracağız. Eğer evinizden dial-up bağlanan bir kullanıcı iseniz sisteminizde dahili bir dnscache kurabilir ve `/etc/resolv.conf` dosyanıza `127.0.0.1` adresini yazabilirsiniz. Dahili denmesinin sebebi `127.0.0.1` arayüzüne kurulması ve dnscache'in kendini o ip'nin 53. portuna bağlamasıdır. Haricinin farkı ise sadece dnscache'in `127.0.0.1`'den farklı ve diğer makinaların ulaşabileceği bir IP olmasıdır.

Ek olarak `dnscache` kurulmadan önce hangi kullanıcı olarak çalışacağı ve hangi kullanıcı olarak günlük kayıtlarının tutulacağı belirtilmelidir. Genel olarak sisteminize giriş yapması ihtimali olmayan (kabuğu `/bin/false` olan) iki kullanıcı ekleyin. Bunlardan biri `dnscache` diğeride günlük kayıtlarını hesap olan `dnslog` olabilir. Bir de programımızın çalışacağı dizin belirlenmelidir. Genel olarak `/etc/dnscachex` (x= external → harici) kullanılmaktadır. Şimdi yazılımı kurabiliriz.

### Harici dnscache

```
# dnscache-conf dnscache dnslog /etc/dnscachex 192.168.1.1
# ln -s /etc/dnscachex /service
# echo "192.168.1.1" > /etc/resolv.conf
```

İlk satırın açılımı:

`dnscache-conf` *çalıştırma-kullanıcısı günlük-kayıd-kullanıcısı ip-adresi*  
şeklinde. İkinci satır ise `dnscache`'i açılıştta çalıştırmak içindir.

### Dahili dnscache

```
# dnscache-conf dnscache dnslog /etc/dnscache
# ln -s /etc/dnscache /service
# echo "127.0.0.1" > /etc/resolv.conf
# touch /etc/dnscachex/root/ip/193.140.1
```

Dikkat ederseniz ilk satırda IP belirtmedik. Öntanımlı değeri `127.0.0.1`'dir

Öntanımlı olarak `dnscache` bütün sorguları reddeder. Son satırda `193.140.1` dosyası ile ağ için sorgu yapılmasına izin veriyoruz.

`dnscache`'in çalışıp çalışmadığını sorgulamak için `dnsqr` komutu kullanılabilir. Yapılan sorgularda önemli olan satır `answer` satırıdır.

```
# dnsqr a yildiz.edu.tr
1 yildiz.edu.tr:
166 bytes, 1+1+3+3 records, response, noerror
query: 1 yildiz.edu.tr
answer: yildiz.edu.tr 30218 A 193.140.1.1 ①
authority: yildiz.edu.tr 86391 NS ana.cc.yildiz.edu.tr
authority: yildiz.edu.tr 86391 NS ns.ulak.net.tr
authority: yildiz.edu.tr 86391 NS vm4381.ce.yildiz.edu.tr
additional: ana.cc.yildiz.edu.tr 86391 A 193.140.1.1
```

```
additional: ns.ulak.net.tr 62760 A 193.140.83.251
additional: vm4381.ce.yildiz.edu.tr 86391 A 193.140.2.1
```

① `answer` ile başlayan kısım bizim aradığımız cevaptır.

Artık elimizde çalışan bir çözümleyici var. Ek olarak istersek bellek büyüklüğünü 100MB ram harcayacak şekilde ayarlayabiliriz. Bunun için:

```
# echo 100000000 > /etc/dnscachex/env/CACHESIZE
# echo 104857600 > /etc/dnscachex/env/DATALIMIT
```

yapılması gerekir. Tabii bundan sonra `svc -t /service/dnscache` ile programımızı yeniden başlatmak gerekecektir.

Şimdi sıra alanımız için yetkili isim sunucu olacak olan `tiny-dns` programını kurmaya geldi.

## 5. Tiny-dns Kurulumu

`Tiny-dns` yukarıda da anlatıldığı üzere bizim alanımız için yetkili isim sunucu olacak. (Aynı zamanda yetkili tersinir isim sunucumuz olacak – ipv4—>hostname...)

Önce sistemimize `tiny-dns` kullanıcıını (`tinydns`) ekleyelim. Ayrıca sistem günlük kayıtlarını tutması için `dnstlog` kullanıcısı da eğer daha önce `dnscache` kurulmadıysa eklenmelidir.



### Dikkat

Dikkat edilmesi gereken bir nokta, `dnscache` ile `tinydns` aynı IP'yi dinleyemez. Bu yüzden `tiny-dns`'i 192.168.1.2'nin 53. portuna bağlayacağız. (IP-ALIASING artık bütün \*NIX'lerde vardır). Eğer elinizde sadece bir Internet IP adresi varsa bu IP'yi `tiny-dns` için kullanın ve yerel 192.168.1.1 IP'sini de `dnscache` için kullanın.

Tiny-dns yapılandırması

```
# tinydns-conf tinydns dnstlog /etc/tinydns 192.168.1.2 {ya da bir Internet IP'si}
# ln -s /etc/tinydns /service {5 saniye sonra çalışır olacak}
```

Sıra `tinydns`'e makina ismi ve tersinir kayıtların girilmesine geldi. Bu aşamada iki seçeneğiniz var. Ya `data` dosyasını elle düzenleyeceksiniz ya da `/etc/tinydns/root` dizini altında bulunan `add*` betiklerini kullanacaksınız. Burada bizim tavsiyemiz `data` dosyasına girdilerin `add*` betikleri yardımıyla girilmesi yönünde olacaktır. Böylece her ne kadar `data` dosyasının yapılandırması `bind`'a göre son derece kolay ve anlaşılır olsa da ufak hatalar yapılabilme olasılığının olmasıdır. Fakat `add*` betikleri ile bu baş ağrıtıcı hataların oluşturduğu problemlerden kurtulmuş olursunuz.

Sırasıyla `yildiz.edu.tr` için dns kayıtlarını girelim ve yetkili isim sunucumuzu kuralım.

```
# cd /etc/tinydns/root
# ./add-ns yildiz.edu.tr 193.140.1.1 ①
# ./add-ns 1.140.193.in-addr-arpa 193.140.1.1 ②
# ./add-host ogis.yildiz.edu.tr 193.140.1.3 ③
# ./add-alias www2.yildiz.edu.tr 193.140.1.3 ④
# ./add-mx yildiz.edu.tr 193.140.1.1 ⑤
# ./add-mx yildiz.edu.tr 193.140.1.5 ⑥
# ./add-childns ece.yildiz.edu.tr 193.140.1.15 ⑦
# make ⑧
```

① `*.yildiz.edu.tr` için yetkili isim sunucusu 193.140.1.1 olacak; SOA

- ② tersinir yetkili isim sunucusu 193.140.1.1 olacak
- ③ Sanırım bayağı zor anlaşılacak birşey :)
- ④ DAHA ÖNCEDEN TANIMLANMIŞ bir IP'ye takma isim ekliyoruz
- ⑤ bir numaralı mx (eposta sunucusu) olarak 193.140.1.1 ekleniyor
- ⑥ ikinci sırada gelen yildiz.edu.tr eposta sunucusu
- ⑦ alt alanlar için isim sunucusu
- ⑧ `tindns-data` komutu çalıştırılır

Bu noktada önemli birşey var. `add-ns` ve `add-mx` betiklerini kullanırken dikkat ederseniz isim sunucuların makina isimlerini vermedik. Bu iki betik öntanımlı olarak `a.ns ... b.ns` makina isimlerini atarlar. Eğer Internic'te ya da daha doğru ifadesiyle `registrar`'ınızda isim sunucusu makina ismi olarak farklı bir isim olabilir. İşte bu noktada iki makina ismi de aynı olmalıdır. `add-ns` ve `add-mx` yine aynı dizinde, `/etc/tinydns/root/` dizini altında bulunan `data` dosyasını düzenler. Burada metin düzenleyicinizle bu dosyayı açıp gerekli değişiklikleri yapmalısınız. Örnek olarak `ulak.net.tr`'de `yildiz.edu.tr` isim sunucusunun makina ismi `ana.cc.yildiz.edu.tr` olarak tanımlıdır. `data` dosyasında bu değişiklik yapılmalıdır.

Sıra geldi `data` dosyasındaki bilgileri `data.cdb` olarak yani `tinydns`'in yapılandırma dosyası olarak kaydetmeye. Yine aynı dizinde dikkat ederseniz bir `Makefile` vardır. Yapılması gereken tek şey bu dizinde `make` komutunu çalıştırmaktır. `Make` komutunun yazılması ile birlikte `tiny-dns` hizmet vermeye hazır olacaktır. `Make` komutu `Makefile` dosyasından da anlaşılacağı üzere `tinydns-data` programını kullanarak `data.cdb`'yi oluşturur. Kısacası `tinydns-data`, `data` dosyasını okur ve `tinydns`'in hızlı bir şekilde ulaşabileceği `data.cdb` dosyasını oluşturur. Ayrıca `tinydns-data` otomatik olarak `data.cdb`'yi günceller. Böylece `data` dosyasında yapılan bir değişiklik `tinydns`'i bir daha başlatmadan doğrudan `data.cdb` dosyasını günceller. `data.cdb` güncellenirken sistem çökse bile sistem yeniden başladığında `data.cdb` dosyasında bir problem yaşamaz. Tam bir ya hep ya hiç mantığıyla ya da veri aktarımı mantığı ile çalışır.

Aşağıda `data` dosyasının içeriği verilmiştir. Bu dosyayı üstte `add-*` betikleri ile yapılandırabileceğiniz gibi elle de yapılandırabilirsiniz.

```
.yildiz.edu.tr:193.140.1.1:a:259200 ①
.1.140.193.in-addr.arpa:193.140.1.1:a:259200
=www.yildiz.edu.tr:193.140.1.1:86400 ②
.1.140.193.in-addr.arpa:193.140.1.1:a:259200 ③
=ogis.yildiz.edu.tr:193.140.1.3:86400 ④
+www2.yildiz.edu.tr:193.140.1.3:86400 ⑤
@yildiz.edu.tr:193.140.1.3:a::86400 ⑥
@yildiz.edu.tr:193.140.1.5:b::86400 ⑦
&ece.yildiz.edu.tr:193.140.1.15:86400 ⑧
```

- ① `yildiz.edu.tr` için yetkili isim sunucusu nokta ile başlıyor
- ② `www.yildiz.edu.tr` için host kaydı
- ③ `*.1.140.193` için yetkili isim sunucusu
- ④ `ogis.yildiz.edu.tr` için makina kaydı
- ⑤ Bir takma ad (alias) kaydı. Takma ad kayıtları `+` ile başlar.
- ⑥ `yildiz.edu.tr` için mx kaydı. Öncelikler harflerle ifade ediliyor.



⑦ yildiz.edu.tr için ikinci mx kaydı. @ ile başlıyor.

⑧ ece.yildiz.edu.tr için gelen sorguları 193.140.1.15 IP'sine yönlendir.

Bu arada istenirse `daemontools` paketi ile gelen `svstat` ile `tinydns` çalışıyor mu çalışmıyor mu kontrol edilebilir.

```
# svstat /service/tinydns
```

Ya da günlük kayıt dosyalarına bakabiliriz.

```
# tail -f /service/tinydns/log/current
```

ile gerçek zamanlı olarak işlemler gözlenebilir.

Bir de `tinydns`'in verisini kontrol edelim. Bunun için `dnscache` için kullandığımız komuta benzer bir komut olan `dnsq` komutunu kullanacağız.

```
# dnsq a yildiz.edu.tr 193.140.1.1
1 yildiz.edu.tr:
127 bytes, 1+1+2+2 records, response, authoritative, weird ra, noerror
query: 1 yildiz.edu.tr
answer: yildiz.edu.tr 86400 A 193.140.1.1 {Önemli olan kısım burası}
authority: yildiz.edu.tr 86400 NS smurf.cc.yildiz.edu.tr
authority: yildiz.edu.tr 86400 NS bigblue.ce.yildiz.edu.tr
additional: smurf.cc.yildiz.edu.tr 86400 A 193.140.1.1
additional: bigblue.ce.yildiz.edu.tr 86400 A 193.140.2.1
```



#### Dikkat

`tinydns`'i sorgulamak için `dnsq`

`dnscache`'i sorgulamak için `dnsqr` kullanılır.

## 6. /service/tinydns/root/data Dosyasının Biçemi

Yukarıda da belirtildiği gibi `data` dosyasından `tinydns`'in okuyabileceği biçime (`data.cdb`) dönüştürülür. Peki `data` dosyasının biçemi nasıldır:

```
.fqdn:ip:x:ttl:timestamp:lo
```

Teker teker yukarıdaki ifadenin ne anlamlara geldiğini açıklayayım.

#### .fqdn

Girdiyi simgeler. Bu girdi yetkili bir isim sunucusunun başlangıcı (SOA= Start Of Authority) olabilir. Nitekim `data` dosyasında "." ile başlayanlar SOA kaydını belirtir. Üstteki örneğimize bakarsak en üst iki satır "." ile başlamaktadır. Bunlar SOA'ları oluşturur. "." dan başka karakterler de vardır:

- . ile başlayanlar SOA'yı belirtir. Mutlaka olmalıdır.
- @ ile başlayan satır bunun bir mx girdisi olduğunu gösterir.
- = ile başlayanlar A kayıtları için kullanılır. Aynı zamanda `tinydns-data` ters kayıtlarından olan PTR girdisi içindir.
- + ile başlayanlar add-alias betiği oluşturulur. Yine bu işaret `tinydns-data`'ya A kaydı oluşturmasını söyler, fakat PTR kaydı girilmez. Eğer aynı IP için iki tane = kullanırsanız hata alırsınız (`data` dosyasını elle değiştirerek ya da `add-host` ile).

- & ile başlayan satırlar alt dns sunucusu girmek içindir. Mesela `elek.yildiz.edu.tr` 'nin kendi dns sunucusunun olmasını istiyorsunuz. Bunun için `add-childdns` komutunu kullanırsınız.

### ip

Girdisi yapılacak fqdn'nin çözümleneceği IP'yi belirler. Mesela:

```
&ece.yildiz.edu.tr:193.140.1.15:86400
```

ifadesindeki ikinci segman IP'dir ve kendisinden önce gelen ifadenin (`&ece.yildiz.edu.tr`) IP'sini belirler.

### x

mx ve ns kayıtlarında işe yaramaktadır. Bu önceliği belirlemek için kullanılır ve sorgulara buna göre verilir. Şöyleki eğer buraya "a" ya da "b" girerseniz `dnsq` ile size dönecek cevap eğer ns kaydı söz konusu ise `a.ns.yildiz.edu.tr` olur. Eğer girilen değer b ise bu ikincil isim sunucusunu gösterir ve `b.ns.yildiz.edu.tr` olarak gösterilir. Yine mx kayıtları içinde `a.mx.yildiz.edu.tr` ve `b.mx.yildiz.edu.tr` kullanılır. Eğer bu tarz gösterim yapmak istemiyorsanız. Bu kısma doğrudan `"dns1.yildiz.edu.tr"` ya da `"dns2.yildiz.edu.tr"` girebilirsiniz. Ya da aynı şekilde `"mx1.yildiz.edu.tr"` ve de `"mx2.yildiz.edu.tr"`.

Aklınıza şu soru gelebilir. Eğer bu kısma ben kendim makina isimlerimi girersem öncelik sırası ne olacak. Burada ilk giren ilk sunulur ilkesi esas alınır. Yani dosyada hangisi üstte ise önceliğe o sahiptir.

### ttl

Kurduğumuz yetkili isim sunucuyu sorgulayan sunucuların belleklerinde bu girdiyi kaç saniye(\*\*) tutacağını belirler. ns ve mx sunucular için burada öntanımlı değer 3 gün (259200 saniye) iken diğer girdiler için 1 gündür (86400 saniye).

### timestamp

Adındanda anlaşılacağı üzere dosyanın ne zaman değiştirildiğini gösterir. " `add-*` " scriptleri bu kısma dokunmazlar.

### lo

Bu kısım aşağıda da anlatıldığı üzere "Değişik istemcileri değişik sunuculara nasıl yönlendirebilirim? " sorusunun cevabı olmaktadır.

EK: 3 kısımda anlatılanları somutlaştırmak için örnek bir data dosyasından alıntı yapayım:

```
.yildiz.edu.tr:193.140.1.1:dns1.yildiz.edu.tr:259200
```

Böyle bir girdide `tinydns-data` komutu iki şey yapar. İlk başta `dns1.yildiz.edu.tr` için `193.140.1.1`'i gösteren A kaydı oluşturulur. Daha sonra bunun SOA olduğunu belirtilen kısım `data.cdb`'ye yazılır.

```
@yildiz.edu.tr:193.140.1.3:mx1.yildiz.edu.tr::86400
```

Bu örnekte ise ilk başta A kaydı sonra da mx kaydı oluşturulur.

## 7. Bölge (zone) Transferi

Bölge (Zone) Transferi Nedir?

Asıl ya da birincil DNS sunucunuzda yaptığınız değişiklikleri ikincil ya da yardımcı sunucularınıza kopyalamanız gerekmektedir. Aslında bunun için bir çok değişik araç mevcuttur. Bunun için öntanımlı kabuğu ssh olan `rsync` bile kullanılabilir. Böylece dosyanın tamamının kopyalanması yerine artan dosyaların sondaki ekleri kopyalanabilir. Ya da isterseniz doğrudan `data.cdb` dosyasını elle kopyalayabilirsiniz.

Fakat bu iş için en akli başında çözüm `axfrdns` kullanmaktır. Bunun için öncellikle:

1. DNS sunucusunun hesap ismi: "axfrdns". Diğer bütün djbDNS programlarında olduğu gibi program kendini bu hesap adı altında belirtilen dizine chroot yapar. İşte bu djbDNS programını güvenli kılan en önemli etkenlerden biridir.
2. DNS günlük kaydı kullanıcısı: dnslog
3. axfrdns'in çalışacağı dizin /etc/axfrdns
4. tinydns dizini: /etc/tinydns

Sonra sırasıyla:

```
# axfrdns-conf axfrdns dnslog /etc/axfrdns /etc/tinydns 193.140.1.1 ①
# cd /etc/axfrdns
# echo "193.140.1.:allow,AXFR=\"yildiz.edu.tr/1.140.193.in-addr.arpa\" > \
/etc/axfrdns/tcp ②
# make
# ln -s /etc/axfrdns /service ③
```

- ① Bu ip tinydns ip'si ile aynı olabilir.Çünkü axfrdns zone transferi için 512 bytetan büyük dosyaları transfer eder bu yüzden tcp kullanılır fakat tinydns udp kullanır.
- ② 193.140.1'deki makinalara reverse ve normal zone'larının transferine izin ver.
- ③ **axfrdns**'i çalıştıralım.

## 8. axfrdns-get (Bölge Transfer İstemcisi)

Eğer birincil sunucu olarak bir BIND sunucu kullanacaksanız ikincil isim sunucu olarak axfr-get programını kullanmalısınız. Ya da diğer bir birincil tinydns sunucusundan bölge transferinde bulunmak için kullanılabilir. Eğer iki sunucu da (birincil ve ikincil) tinydns ise scp ya da rsync ile doğrudan ana sunucudan data.cdb'yi ikincil sunucunun /etc/tinydns/root/ dizinine kopyalayabilirsiniz. Eğer rsync hususunda bir bilgi almak isterseniz <http://www.freebsdidiary.org/> adresini ziyaret edebilir ya da "man rsync" ile kılavuz sayfasından bilgi alabilirsiniz.

Axfrdns-get kısaca şöyle çalışır:

```
# tcpclient yetkili-isim-sunucu-IP 53 axfrdns-get yildiz.edu.tr fn fn.tmp
```

Burada eğer axfrdns-get çeşitli sorguları yaparak alan için birincil sunucuyu bulacak ve eğer birincil sunucuda bölge transferi için axfrdns-get komutunun çalıştığı sunucuya izin verilmişse birincil sunucudan çektiği yildiz.edu.tr hakkındaki bilgileri fn.tmp olarak yazacaktır. Eğer bölge transfer işlemi başarı ile tamamlanırsa dosya adı otomatik olarak fn olarak değiştirilecektir. Bu fn dosyası da tinydns-data biçiminde olacaktır. Yani daha önce gördüğümüz data dosyası ile aynı biçimde olacaktır.

Genel olarak axfrdns-get programının getirdiği dosyalarda aynı kayda ait birden fazla girdi olacaktır. Dosyayı "sort -u" komutu ile sıralayabilirsiniz.



### Uyarı

axfrdns-get için daha önceden tcpclient programını kurmuş olmanız gerekmektedir. Zaten sisteminiz tcpserver kurulu ise tcpclient otomatik gelecektir.

## 9. DjbdNS Komutları

Şu an kadar anlatılanlar neyin nasıl yapılacağı hususunda genel bir bilgi vermek içindi. Bu bölümde anlatılacak olanlar djbDNS ile gelen programların (hepsinin) teker teker kısaca bir açıklaması olacak. Djbdns ile gelen bir çok komut var. Bunların aklınızda biraz daha kolay yer etmesi ve bu belgenin bir başvuru belgesi kabul edilmesi için bu tarz bir uygulama yapıyorum.

## 9.1. dnscache

Kısaca çözümleyici olarak kullanılmaktadır. Çözülen her isteği aynı zamanda bellekleyerek cevap verme süresinin azalmasını sağlar. Ağ ortamından gelen isteklere cevap verecek şekilde ayarlanabileceği gibi yerel çözümleyici olarak kullanılabilir.

## 9.2. tinydns

Yetkili bir isim sunucusudur. UDP altında çalışır. Örnek vermek gerekirse yildiz.edu.tr alanı için gelebilecek tüm sorgulara cevap verecek şekilde ayarlanabilir. 512 byte ve üzeri isteklere cevap vermez. Bunun sebebi sadece UDP altında çalışması ve DNS yapılandırmasının bu şekilde olmasıdır.

## 9.3. walldns

WallDNS tersinir bir DNS duvarıdır. Yani sisteminiz gerisindeki Internet IP'ye sahip makinaların isimlerini saklamakta kullanılabilir. Güvenlik duvarı mantığı ile örtüşür. Bazı ftp sunucuları kendine gelen istekleri kabul etmek için mutlaka ters kaydın o makina için tanımlı olmasını isterler. Bu durumlar için walldns kullanılabilir.

## 9.4. dnsq

tinydns gibi yetkili isim sunucularını sorgulamak için kullanılır.

```
# dnsq type FQDN server
# dnsq MX yildiz.edu.tr 193.140.1.1
```

193.140.1.1 sunucusuna yildiz.edu.tr için kayıtlı mx kaydını sorar.

## 9.5. dnsqr

Bu komut dnsq'ya benzer fakat yetkili isim sunucusu sorgulaması yerine dnscache gibi depoları sorgulamaya yarar.

```
#dnqr ns http://www.yildiz.edu.tr/
```

Depoya yildiz.edu.tr'nin ns kaydını sorar.

## 9.6. dnsfilter

Bu komut dnsname gibidir. Fakat aynı anda çoklu sorgulama yapabilir. dnsname sadece bulduğu ilk girdiyi listeler.

```
# dnsfilter
193.140.1.1                                {Bizim girdimiz}
193.140.1.1=smurf.cc.yildiz.edu.tr
193.140.1.3                                {Bizim girdimiz}
193.140.1.3=ogis.cc.yildiz.edu.tr
```

## 9.7. axfrdns

Bölge transferi için kullanılır. Çeşitli ikincil isim sunucular için bölge transfer sunucusu olarak vazife yapar. Tinydns'in çalıştığı IP'de çalışabilir. TCP kullanır.

## 9.8. dnstrace

Belirtilen kök isim sunucusundan başlayarak ( r ) bir alan altında ( fqdn ) istenenen tipte ( t ) sorguları bulur ve ilişki kurduğu bütün sunucuları standart çıktıya yazar. Fakat karmaşık bir biçimde.

```
# dnstrace t fqdn r
# dnstrace any www.yildiz.edu.tr a.root-servers.net > yildiz &
```

## 9.9. dnstracesort

dnstrace komutunun çıktısını okuyabileceğimiz şekilde listeler.

```
# dnstracesort < yildiz | less
```

Ya da doğrudan:

```
# dnstrace any www.yildiz.edu.tr a.root-servers.net | dnstracesort | less
```

## 9.10. axfr-get

İkincil sunucularda kullanılmak üzere tasarlanmıştır. Birincil sunucudan bölge transferinde kullanılır.

Ayrıca birkaç da betik vardır: dnsmx, dnsip, dnsname

# 10. Yardımcı Bilgiler

Bu bölümde şahsımın ilgisini çeken birkaç noktayı anlatacağım:

## 10.1. Aynı veriyi içeren 5 web sunucusu için yük dengelemesini nasıl yaparım?

Aslında dns tabanlı yük dengelemesine tam manasıyla sıcak bakmasam da djbDNS ile yük dengelemesi yapılabilir. Eğer Linux altında gerçek bir yük dengelemesi ya da Yüksek Kullanılabilirlik arıyorsanız <http://www.linux-vs.org/> ya da adreslerine bakın derim. Neyse biz konumuza dönelim. Amacımıza ulaşmak için TTL ile oynayacağız. Her bir www sunucumuzun girdisi için 5 saniye bir TTL tanımlayacağız. Böylece karşı tarafta depolanan sorguların yaşam süresi 5 saniye olacak. Gelen her bir istek için djbDNS'te döner turnuva yaparak bu 5 sunucu arasında tercihini yaparak karşı taraftaki çözümleyiciye bildirecektir. `/etc/tinydns/root/data` dosyasına

```
+www.yildiz.edu.tr:193.140.1.1:5
+www.yildiz.edu.tr:193.140.1.2:5
+www.yildiz.edu.tr:193.140.1.1:5
.....
```

yazılır. `/etc/tinydns/root/` dizininde `make` komutu çalıştırarak `tinydns-data` komutunun `data` dosyasındaki bilgileri `data.cdb` gibi cdb biçiminde saklaması sağlanır.

## 10.2. Değişik istemcileri değişik sunuculara nasıl yönlendirebilirim?

Diyelim ki üniversiteniz ya da şirketiniz içinden gelen sorgulara yani intranetinizden gelen sorgulara `http://www.yildiz.edu.tr/` için 193.140.1.1 Internet'ten gelen `http://www.yildiz.edu.tr/` sorguları için

192.193.140.1.2...5 arası cevapların verilmesini istiyorsunuz. Bunun için `/etc/tinydns/root/data` dosyasına farklı iki konum tanımlarız. IN ve EX gibi 1 ya da 2 byte uzunluğunda (maximum 2 byte uzunluğunda) tanımlamalar yaparız. Şöyleki:

```
%IN:193.140.1
%EX
+www.yildiz.edu.tr:193.140.1.1:5::IN
+www.yildiz.edu.tr:193.140.1.2:5::EX
+www.yildiz.edu.tr:193.140.1.3:5::EX
...
```

### 10.3. Aynı ağda bulunun bir dnscache o ağ için tanımlı yetkili isim sunucusunu nasıl bulacak?

Normalde aşağıdaki bir düzenleme yapılmazsa yerel olarak bulunabilecek bir cevap için bütün Internet'i gezmesi gerekecektir. Bu durumu önlemek için:

```
# cd /etc/dnscachex/root/servers
# echo "193.140.1.1" > yildiz.edu.tr
# echo "193.140.1.1" > 1.140.193.in-addr.arpa
```

ile bu sorunu çözebilirsiniz.

### 10.4. djbDNS çalışmadı...

Bu kısım çok önemli (!!!). Diyelimki djbDNS'i kurdunuz ve herşeyin tam olarak çalışıp çalışmadığından emin olmak için sisteminizi yeniden başlattınız. Fakat o ne... DjbDNS çalışmadı!!! ve şu şekil bir hata veriyor:

```
svscan: warning: unable to start supervise dnscache: file does not exist
```

Burada bulunamayan ve svscan'ın dnscache'i çalışmamasını sağlayan `/usr/local/bin` altında `daemontools` komutlarının bulunmaması.

Buradaki çözüm: `/usr/local/etc/rc.d/djbDNS.sh` gibi bir dosya oluşturup, `chmod 755 djbDNS.sh` ile çalıştırılabilir hale getirip şu satırları `djbDNS.sh`'a eklemektir.

```
#!/bin/sh
export PATH="$PATH:/usr/local/bin"
svscan /service &
```

## 11. Kaynaklar

### DJB DNS SOURCE

<http://cr.yp.to/>

### DJB DNS HOMEPAGE

<http://www.djbDNS.org/>

<http://djbDNS.enderunix.org/> (Resmi olarak djbDNS.org'un yansısını tutmaktayız.)

Ayrıca bu sayfalarda djbDNS için yapılmış birkaç yama bulabilirsiniz. Örnek olarak bir deponun dnscache ile birden çok IP'ye bağlanması verilebilir.

### DJB DNS FAQ

<http://cr.yp.to/djbDNS/faq/>

### DjbDNS ve BIND Karşılaştırması

<http://cr.yp.to/djbdns/ad/easeofuse.html>

Bu belgeye resmi djb sitesi vasıtasıyla da ulaşabilirsiniz: <http://www.djbdns.org>

### DjbDNS'in tinydns paketini web'den yönetmek için python ile yazılmış bir araç

<http://www.linuxis.net/twa>

### Gerrit Pape'in yazdığı djbdns kılavuz sayfaları

Djbdns ile manual page gelmemektedir: <ftp://ftp.innominat.org/gpa/djb/>

### dnscache için döner turnuva yaması

Bu yama dnscache içindir. tinydns için döner turnuva desteği paketle gelmektedir.

<http://www.legend.co.uk/djb/dns/round-robin.patch>

## 12. Yasal Açıklamalar

### 12.1. Telif Hakkı ve Lisans

Bu belgenin, *djbDNS Kurulumu NASIL* 1.0 sürümünün **telif hakkı © 2001 Ömer Faruk Şen**'e aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan **GNU Genel Kamu Lisansı**<sup>(B18)</sup>'nin 2. ya da daha sonraki sürümünün koşullarına bağlı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın özgün kopyasını <http://www.gnu.org/copyleft/gpl.html> adresinde bulabilirsiniz.

Linux, Linus Torvalds adına kayıtlı bir ticarî isimdir.

### 12.2. Feragatname

Bu belgedeki bilgilerin kullanımından doğacak sorumluluklar, ve olası zararlardan belge yazarı sorumlu tutulamaz. Bu belgedeki bilgileri uygulama sorumluluğu uygulayana aittir.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim bir ticarî isim yada kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.

## Notlar

Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.

<sup>(B18)</sup> [../howto/gpl.pdf](http://howto/gpl.pdf)

---

Bu dosya (djbdns-kurulumu-nasil.pdf), belgenin XML biçiminin T<sub>E</sub>XLive ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

30 Ocak 2007