txdyn - DYNDNS with djbdns

Frank W. Bergmann, tuxad.com [http://www.tuxad.com]
<txdyn.WANTS.N0-5P4M@tuxad.STOP-SPAM.com>

2007-03-28

Revision History

Revision 1.0	2007-03-28	FB
	Initial release.	

Abstract

txdyn is a *DYNDNS* solution based on djbdns (*[DJBDNS]*) and daemontools (*[DAEMONT]*). It mainly contains a shell script which reads the (DYN)DNS data from a filesystem hierarchy and creates a data file for djbdns/tinydns.

Table of Contents

1. About this document

1.1. Disclaimer

No liability for the contents of this document and the usage of the described software can be accepted. Use the software, concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility.

in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements.

1.2. Copyright and License

1.3. Acknowledgments

For their software and the inspiration I received by their work I would like to thank D. J. Bernstein (djb) and Felix Leitner (fefe).

1.4. Conventions used in this document

The following typographic and usage conventions occur in this text:

Table 1. Typographic and usage conventions

Text type	**Meaning**
“Quoted text”	Quotes from people, quoted computer output.
`terminal view`	Literal computer input and output captured from the terminal.
command	Name of a command that can be entered on the command line.
`option`	Option to a command, as in “the `-a` option to the **ls** command”.
`parameter`	Parameter to a command, as in “read **man** ***`ls`***”.
`filename`	Name of a file or directory, for example “Change to the `/usr/bin` directory.”
The author [http://www.tuxad.com]	Click-able link to an external web resource.
GNU FDL (*[GFDL]*)	A Reference to an external source
DYNDNS	A glossary keyword

Thanks to Machtelt “Tille” Garrels for this list of conventions.

2. About txdyn and djbdns

2.1. txdyn, a DYNDNS solution for djbdns

Running my own nameservers (in one subnet) and having access to a server in a different subnet I thought about a *DYNDNS* solution. For a long time I used software by D. J. Bernstein including his software suite djbdns (*[DJBDNS]*) with its easy-to-use, small, fast, secure and reliable DNS-server **tinydns**. Therefore I wanted a DYNDNS solution with **tinydns** and as simple as **tinydns**.

To realize DYNDNS with **tinydns** you have to (continously) alter its database, the file `data`. Using **ed**, **sed**, **perl** or something similar would be a chaotic solution, no easy task to implement and can cause data loss. It seemed more easy to create the file `data` periodically. But `data` itself acts as the database for **tinydns** and what database should be the database for this?

Software engineers like DJB have shown that the filesystem and the files can be a easy-to-use database. With this idea in mind I started to write the main script **create-data.sh**. This script collects data from a file hierarchy (file contents and filenames) and creates a new `data` file. It's easy to use a webserver script to write DYNDNS data in this file hierarchy, it's easy to get an overview or search in the nameserver data and it's very easy i.e. to move a www-domain from one IP-address zo a different address:

```
[me@dns ip] mv 1.2.3.4/aliases/mydom.org 1.2.3.5/aliases/
```

.

2.2. About djbdns

The following text is taken from http://cr.yp.to/:

It works for Lycos. It works for citysearch.com. It works for pobox.com. It works for 1.85 million more .com's. It works for several of the Internet's largest domain-hosting companies: directNIC, MyDomain/NamesDirect, Interland, Dotster, Easyspace, Namezero, Netfirms, and Rackspace Managed Hosting. It'll work for you too.

djbdns is a collection of Domain Name System tools. It includes software for all the fundamental DNS operations:

- DNS cache: finding addresses of Internet hosts. When a browser wants to contact www.hotwired.com, it first asks a DNS cache, such as djbdns's dnscache, to find the IP address of www.hotwired.com. Internet service providers run dnscache to find IP addresses requested by their customers. If you're running a home computer or a workstation, you can run your own dnscache to speed up your web browsing.
- DNS server: publishing addresses of Internet hosts. The IP address of www.hotwired.com is published by HotWired's DNS servers. djbdns includes a general-purpose DNS server, tinydns; network administrators run tinydns to publish the IP addresses of their computers. djbdns also includes special-purpose servers for publishing DNS walls and RBLs.
- DNS client: talking to a DNS cache. djbdns includes a DNS client C library and several command-line DNS client utilities. Programmers use these tools to send requests to DNS caches.

djbdns also includes several DNS debugging tools, notably dnstrace, which administrators use to diagnose misconfigured remote servers.

The djbdns package includes three servers that publish local host information: tinydns, walldns, and rbldns. Every aspect of configuration was rethought from the perspective of an overworked administrator who has better things to do than play with DNS.

tinydns handles basic DNS service. The tinydns-data file format combines the flexibility of zone files with the convenience of modern zone-building tools. Host information is stored in one file. PTR records are handled automatically. Changes can be scheduled in advance, with TTLs handled automatically.

tinydns has several load-balancing features. It automatically selects a random set of 8 servers from a cluster of any size. It allows easy removal of dead servers by external monitoring tools. It also supports client differentiation, checking the client's IP address and choosing one of several clusters accordingly.

Databases for tinydns and rbldns are compiled into cdb format. The servers start responding immediately, even if the database is a gigabyte or more. (In contrast: BIND cannot answer questions until it has loaded all your data into memory.)

One site reported tinydns answering 6000 queries per second on a dual Pentium III-1000 using 40%

of one CPU. That's real queries, not peak performance in lab tests.

While a new database is being compiled, the servers continue to answer queries from the old database. There is no gap in DNS service when the new database is finished. The old database is left in place if anything goes wrong.

Database compilation is very fast. One site reported tinydns-data taking under a minute on a Pentium III-550 to create a 350-megabyte data.cdb covering almost 300000 domains.

2.3. About daemontools

The following text is taken from http://cr.yp.to/:

daemontools (*[DAEMONT]*) is a collection of tools for managing UNIX services.

supervise monitors a service. It starts the service and restarts the service if it dies. Setting up a new service is easy: all supervise needs is a directory with a run script that runs the service.

multilog saves error messages to one or more logs. It optionally timestamps each line and, for each log, includes or excludes lines matching specified patterns. It automatically rotates logs to limit the amount of disk space used. If the disk fills up, it pauses and tries again, without losing any data.

3. txdyn in detail

txdyn is mainly a script and two optional programs. The script creates a data file for **tinydns**. The optional programs are a small "webserver" and a tool which launches the script periodically by calling **make**.

3.1. The filesystem hierarchy of txdyn

The base directory for the script **create-data.sh** is the root-subdirectory of the tinydns-service:

```
/service/tinydns/root
```

The following description uses file and directory paths relative to this base directory.

`create-data.sh`	the main script
`Makefile`	its main target just calls **./create-data.sh**
`ip/`	the directory holding all nameserver data to create the data file for tinydns
`ip/IP_LIST`	contains an ordered list of all IP addresses resolved by this nameserver
`ip/NS_DOMAIN_ORDER`	ordered list of domains with nameservers (e.g. tuxad.net which has ns1.tuxad.net and ns2.tuxad.net)
`ip/10.1.1.1/`	base config dir for IP-address 10.1.1.1
`ip/10.1.1.1/host`	contains hostname for the host with this IP#
`ip/10.1.1.1/mx/`	directory holding all mx hosts for this IP#
`ip/10.1.1.1/mx/mail.mydom.org`	filename is mx hostname, content specifies domain served by this mx host

`ip/10.1.1.1/mtamark`	DNS MTAMARK [http://mtamark.space.net/], contains 1 if server acts as mail-server
`ip/10.1.1.1/aliases/`	every file in this directory specifies an alias
`ip/10.1.1.1/aliases/www.mydom.org`	an alias entry
`ip/10.1.1.1/ns-soa/`	contains nameservers (structure like mx dir)
`ip/10.1.1.1/ns-soa/ns1.mydom.org`	name is host, content are domains
`ip/DYNDNS/`	holds all domains with dyndns-names
`ip/DYNDNS/user1.mydom.org`	content's first line is IP#, 2nd line is time-to-live in minutes

3.2. The script create-data.sh

The script **create-data.sh** creates a file `./data` suitable for **tinydns** with the information it collects from the directory `./ip` like it's described in before.

If the nameserver data in the `./ip` directory will change frequently then **create-data.sh** must be started periodically in short intervals. This is the case if you want to use DYNDNS. One possible solution is a cronjob, another solution is to use tinydnsupdate which will call the script by **make** more often than a minutely cronjob.

create-data.sh creates a PID-file in the `./tmp` directory to avoid multiple running instances.

create-data.sh does not create a new `data`-file if there were no changes of nameserver data in the `./ip`-directory.

3.3. tinydnsupdate

tinydnsupdate substitutes a cronjob for periodically calling **create-data.sh**. If even a minutely cronjob is not frequently enough for calling **create-data.sh** then this small helper tool can be used.

tinydnsupdate reads the environment variable `TINYDNS_ROOT`, does a **chdir** into this directory and calls **make** forever with breaks of 10 seconds.

Its working directory must be the directory which contains **create-data.sh** and its `Makefile` (default: `/service/tinydns/root`).

tinydnsupdate is designed to run under `daemontools`. You must create a directory for it, containing the tool itself, a daemontools run script and an `env`-directory with the file `TINYDNS_ROOT`. Here's an example:

```
$ cd /service/tinydns/root/tinydnsupdate/
$ ls -l
total 32
dr-xr-xr-x 2 tinydns tinydns 4096 Apr  9 14:22 env
-r-xr-xr-x 1 tinydns tinydns   62 Apr  9 14:22 run
drwx------ 2 tinydns tinydns 4096 Apr  9 14:22 supervise
-r-xr-xr-x 1 tinydns tinydns 1252 Apr  9 14:22 tinydnsupdate
$ cat env/TINYDNS_ROOT
/service/tinydns/root
$ cat run
#!/bin/sh
exec setuidgid tinydns envdir ./env ./tinydnsupdate
```

As the example shows the directory can be set up in the tinydns `root`-directory. This tool must not run with root rights. It should be run as the user who may create the `data`-file. To start it you must

create a symlink in the `/service`-directory to this directory:

```
$ cd /service
$ ln -s tinydns/root/tinydnsupdate
```

3.4. The "webserver" tyrre

3.5. cgi-bin script dyndns

4. Installing from scratch

4.1. Requirements

There are a few requirements to install/run the txdyn software:

- base environment with **bash** and **sed**
- **make** (with pattern rules support)
- **rsync**
- `djbdns` with **tinydns** already configured
- DJB's `daemontools` already configured
- default required, but actually optional:
  - **gcc**
  - **dietlibc**
  - DJB's `ucspi-tcp`

4.2. Quick installation

Here's a sample installation for a quick start.

Example 1. Installation Instructions

1. Unpack the archive:

   ```
   tar xzf txdyn-0.5.tgz
   ```

2. Step into txdyn/src-directory:

   ```
   cd txdyn-0.5/src
   ```

3. Call **make** and specify the IP-addresses of your master and one or more slave nameservers:

   ```
   make MASTER=10.1.1.1 SLAVES="10.2.2.1 10.3.3.1"
   ```

4. Do a **make install** as root:

```
su -c 'make MASTER=10.1.1.1 SLAVES="10.2.2.1 10.3.3.1" install'
```

If this fails and an error about a missing user is displayed then install the user accounts and do it again:

```
su -c 'make install-users'
su -c 'make MASTER=10.1.1.1 SLAVES="10.2.2.1 10.3.3.1" install'
```

5. Configuration

config

6. Using txdyn

using

7. Troubleshooting

If something doesn't work right please read this section.

8. References

[GFDL]	GNU FDL http://www.gnu.org/licenses/fdl.html
[DJBDNS]	http://cr.yp.to/djbdns.html
[DAEMONT]	http://cr.yp.to/daemontools.html
[UCSPI]	http://cr.yp.to/ucspi-tcp.html

9. Glossary

Glossary

DYNDNS

the name of a company and also a commonly used term for dynamic name service:

“ Dynamic DNS is a system which allows the domain name data held in a name server to be updated in real time. The most common use for this is in allowing an Internet domain name to be assigned to a computer with a varying (dynamic) IP address. This makes it possible for other sites on the Internet to establish connections to the machine without needing to track the IP address themselves. A common use is for running server software on a computer that has a dynamic IP address, as is the case with many consumer Internet service providers. ” (Wikipedia)

Index