

TIME SYNCING OVER HTTP

All administrators know how important it is to be keep your whole network running at the same times, it is not only useful because suddenly comparing logs across different devices makes more sense, but certain server setups require a proper time setup on your network. We all know this, but getting NTP (Network Time Protocol) working is something we put off - either because we are lazy or because we are paranoid and do not want to open more ports then needed on our firewall. No matter what the reason, there is an easier alternative - you can sync your time over http, as this port is generally already open. You can do this using *htpdate* which can be found [here](#). Lets take a look..

What is Needed?

Well, you first need the *htpdate* tool, so download it (the version could change)...

```
wget http://www.clevervest.com/http/archive/c/htpdate-0.9.1.tar.gz
```

And you will of course need to be to get out onto the web.

Getting it Setup

Once you have the tarball, it is a fairly simple process..

```
# gzip -d htpdate-0.9.1.tar.gz
# tar -xvf htpdate-0.9.1.tar
# cd htpdate-0.9.1
# make
gcc -Wall -ansi -Os -DDEBUG -o htpdate htpdate.c
# make install
/usr/bin/strip -s htpdate
mkdir -p /usr/bin
/usr/bin/install -c -m 755 htpdate /usr/bin/htpdate
mkdir -p /usr/share/man/man8
/usr/bin/install -c -m 644 htpdate.8.gz /usr/share/man/man8/htpdate.8.gz
```

See, not difficult at all. Now you just need to do a test. You need to choose websites that are near you (in your country) and preferably more than one...

```
# htpdate -d -q www.google.co.uk www.bbc.co.uk
www.google.co.uk      14 Feb 2006 07:45:35 GMT (3.560) => -61
www.bbc.co.uk         14 Feb 2006 07:45:44 GMT (5.678) => -62
#: 2, mean: -61, average: -61.500
Offset -61.500 seconds
```

The *-q* is used to just query the servers and the *-d* is the debug switch. What we were checking was just that the servers we specified were able to provide us with meaningful information.

Finishing the Job

Once you have found the servers you can use and now want to automate the checking, you can either use the builtin *htpdate* daemon mode, or you can use *crontab*. Personally I prefer *crontab*, so you can setup a task something like..

```
25 * * * * /usr/bin/htpdate -d -a -l www.google.co.uk www.bbc.co.uk
```

The *-a* tells the process to adjust the time gently and the *-l* tells the process to log to syslog. The above entry now means my server will check the time every hour at 25 minutes past. Nice and easy.

Final Words

The *htpdate* tool is any very good example of the maxim that there is more than one way to skin a cat. It is small, easy, has options for proxy usage and does the job nicely. As always have fun and learn.