Translation(s): none

Inhaltsverzeichnis

> Dnsmasq is a lightweight, easy to configure, DNS forwarder and DHCP server. It is designed to provide DNS and optionally, DHCP, to a small network. It can serve the names of local machines which are not in the global DNS. The DHCP server integrates with the DNS server and allows machines with DHCP-allocated addresses to appear in the DNS with names configured either in each host or in a central configuration file. Dnsmasq supports static and dynamic DHCP leases and BOOTP/TFTP for network booting of diskless machines (source: from the DebianPkg: package description).

Basic DNS Setup

First things first, let's install the package:

```
apt-get update
apt-get install dnsmasq
```

If your goal was to set up a simple DNS server, you just succeeded. To test it, use your favorite DNS lookup tool pointed at localhost:

```
dig debian.org @localhost
```

or

```
nslookup debian.org localhost
```

By default, DNS is configured to forward all requests to your system's default DNS settings. In case you didn't know, these are stored in the `/etc/resolv.conf` file. See Debian Reference or the DebianMan: resolv.conf(5) man page for more details.

Now, if you want to add some names for your DNS server to resolve for your clients, simply add them to your `/etc/hosts` file.

Choosing Your Interfaces

One you will probably want to do is tell dnsmasq which ethernet interface it can and cannot listen on, as we really don't want it listening on the internet. Around line 69 of the `/etc/dnsmasq.conf` file, you will see:

```
#interface=
```

Uncomment the line and specify which ethernet interface(s) you want it server IPs to. For example, if I want it to listen on eth1 (my DMZ) and eth2 (my local network), then it should look like:

```
interface=eth1
interface=eth2
```

If I didn't edit this line, it would also listen on eth0, my internet connection. I personally wouldn't recommend this, as it gives those evil guys a few doors to try to break into.

Basic DHCP Setup

By default, DHCP is turned off. This is a good thing, as you could bring down whatever network you are connected to if you are not careful.

To enable it, there is at least one line will need to edit in the `/etc/dnsmasq.conf` file. Around line 143, you will see:

```
#dhcp-range=192.168.0.50,192.168.0.150,12h
```

To enable the DHCP server, you will need to give it a range of IP addresses

to hand out. In the example above, this server would hand out 101 address starting at 192.168.0.50 and ending at 192.168.0.150. The last number is how long the DHCP leases are good for. In this example, they would be good for twelve hours.

Since I have two different networks that need DHCP, I'm going to change that line to:

```
dhcp-range=eth1,192.168.100.100,192.168.100.199,4h
dhcp-range=eth2,192.168.200.100,192.168.200.199,4h
```

Notice the "eth1" and "eth2" labels in the lines above? The aren't necessary, but definately help once you start playing with more advanced configurations. It also helps me remember which range is which. Now restart your dnsmasq server, connect up a few clients, and see if they autoconfigure themselves:

```
/etc/init.d/dnsmasq restart
```

Local Caching

Using dnsmasq to cache DNS queries for the local machine is a bit tricky, since all DNS queries from the local machine need to go to dnsmasq, while as the same time, dnsmasq must be configured to forward all those queries to upstream DNS servers.

Do not use this configuration if you use different network (e.g If you use a laptop!)

The DebianMan: dnsmasq(8) man page suggests the following:

> In order to configure dnsmasq to act as cache for the host on which it is running, put "nameserver 127.0.0.1" in /etc/resolv.conf to force local processes to send queries to dnsmasq. Then either specify the upstream servers directly to dnsmasq using --server options or put their addresses real in another file, say /etc/resolv.dnsmasq and run dnsmasq with the -r /etc/resolv.dnsmasq option. This second technique allows for dynamic update of the server addresses by PPP or DHCP.

There is, however, a simpler method; simply ensure that the machine's list of

nameservers contains the line

```
nameserver 127.0.0.1
```

as the *first* line, followed by the upstream nameservers. dnsmasq is smart enough to ignore this line and forward all queries appropriately, while all other applications will send all their queries to dnsmasq.

Exaclty how to do this depends on the method(s) of network configuration in use. If you're manually hardcoding the nameservers (either in `/etc/resolv.conf` or elsewhere, such as a stanza in `/etc/network/interfaces` or in the Wicd GUI), then just add a reference to `127.0.0.1` as the first entry in the list. If you're using DHCP, then instruct your client to prepend `127.0.0.1` to the DHCP servers it receives. E.g., with dhclient, include the line

```
prepend domain-name-servers 127.0.0.1;
```

in the dhclient configuration file (`/etc/dhcp3/dhclient.conf`). [On my Sid system, the default configuration file shipped with the package contains that line, but commented out.]

Note that if you plan to use dnsmasq for the local system only, you should lock it down by adding the line

```
listen-address=127.0.0.1
```

to the dnsmasq configuration file (`/etc/dnsmasq.conf`).

See Also

- http://www.debian.org/doc/manuals/debian-reference/ch05 - Debian Reference: Chapter 5. Network setup
- http://www.thekelleys.org.uk/dnsmasq/doc.html - dnsmasq home page

CategoryNetwork