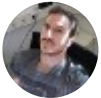




Published in CodeX

You have **1** free member-only story left this month. [Sign up for Medium and get an extra one](#)



Ziga Petek

Follow

Aug 7, 2022 · 4 min read · ✨ · 🎧 Listen



Save



How to Create a Home Network and Reverse Proxy with DnsMASq and Nginx

Make Your Home Network Enjoyable



If you are like me you have a lot of stuff running at home. A media center, a file server, and maybe a few home automation systems,... Remembering all those IPs and ports can be a pain. Here I am going to show you how you can build a DNS server and remote proxy for your home with DnsMASq and Nginx.

Admit it, opening your browser and simply writing an address like “www.google.com” instead of 123.123.123.123 is more convenient. But that does not happen magically. Someone has to map the url to the IP, just like in the olden days' switchboard operators have connected you to the person you wanted to speak to when picking up the phone.

And that someone are DNS servers. They decode the domain name and forward your request to the right IP so you can get the results on the first page of Google (the second page of the results is a myth, do not let them fool you).

If you have a few services running at home, you can build your own DNS server, that will decode custom domains into internal IP addresses. Unfortunately, you will need more than a DNS server to take care of all the different ports. That's why you also need a reverse proxy.

For the DNS server, we will use DNSMASq, a small and versatile DNS server. It will take an URL and first look if it can decode it internally. If not, it forwards it to an external DNS server.

we will use Nginx for our reverse proxy. I already explained how to set up a reverse proxy here:

How to Use Nginx as a Reverse Proxy

Build a Gate Keeper For Your Servers

ziga-petek.medium.com

so I'll only explain in detail how to set up DnsMASq.
So, let's start.

Let us get to work



Photo by [Marissa Grootes](#) on [Unsplash](#)

First, you have to define under which domain all of your computers should be accessible. For the sake of this tutorial let's take *home.lan*.

If the applications you want to access reside on different computers it is probably best if you install DnsMASq on a separate computer. I installed mine on an old raspberry pi I was having lying around.

Before installing DnsMASq you will need to disable systemd-resolved since both use port 53 for running and it would cause complications:

```
1 apt-get update -y
2 sudo systemctl stop systemd-resolved
3 sudo systemctl disable systemd-resolved
```

disable-systemd-resolved.sh hosted with ♥ by GitHub

[view raw](#)

The first line updates your Linux system. That's the standard line you should use before starting any project on Linux.

The second line stops systemd-resolved and the third line disables the service preventing it to run on startup.

Now we are ready to install DnsMASq:

```
1 sudo apt install dnsmasq
2 sudo apt install dnsutils
```

install_dnsmasq.sh hosted with ♥ by GitHub

[view raw](#)

we will also need **DnsUtils**, that's what the second line is for.

To run DnsMASq properly you will need to create a configuration file. Create the file */etc/dnsmasq.d/home.lan* with the following contents:

```
1 bogus-priv
2 domain=home.lan
3 expand-hosts
4 no-hosts
5 domain-needed
6 no-resolv
7 no-poll
8 server=8.8.8.8
9 server=4.4.4.4
10 address=/media.home.lan/127.0.0.1
11 address=/router.home.lan/127.0.0.1
```

home.lan hosted with ❤ by GitHub

[view raw](#)

First of all, why did we create a separate configuration file instead of editing */etc/dnsmasq.conf*? This is a good way of encapsulating configurations for different purposes and you should do that everywhere, not just with DnsMASq.

Some explanation of what the different lines in the configuration file do:

domain-needed

bogus-priv

both lines prevent packets with malformed domain names and packets with private IP addresses from leaving your network. Both should be included.

domain=home.local

local=/home.local

This lines tells DnsMASq what your domain name will be.

no-hosts

This line tells DnsMASq to not read the entries found in the file */etc/hosts*.

no-resolv

imit your name services exclusively to DnsMASq.

no-poll

Tells DnsMASq to not poll */etc/resolv.conf* or other resolv files for changes and re-read them.

server=8.8.8.8

server=8.8.4.4

Those lines use DnsMASq which external DNS servers to use if it can not find the requested domain. In our case we used the old reliable Google DNS servers.

```
address=/media.home.lan/127.0.0.1
```

```
address=/router.home.lan/127.0.0.1
```

Those lines tell DnsMASq where to point the different subdomains to. Since we are going to use NginX for this, we use the IP of the computer where we can find it. And since we are going to install NginX on the same computer the IP is 127.0.0.1.

Now we can restart DnsMASq:

```
1 sudo systemctl restart dnsmasq
```

restart_dnsmasq.sh hosted with ♥ by GitHub

[view raw](#)

Everything should be working by now.

And what now? DnsMASq only handles domains. The ports have to be handled by a different system. We do not want to remember all the ports either, so we install Nginx and configure it as a reverse proxy.

Open in app ↗

Sign up

Sign In



article and do that now:

How to Use Nginx as a Reverse Proxy

Build a Gate Keeper For Your Servers

ziga-petek.medium.com

Conclusion

That's it. By now you should have a working DNS server. Now you only have to configure each device connected to your local network to use the IP of your DNS

server and you should be able to open your applications by using your domain names.

Happy coding.

Related Articles

How to Host a Website With a Dynamic IP And no Open Port

In case your ISP is a jerk

ziga-petek.medium.com

How to Point a Domain Name to Your Server

So people will easier find you

ziga-petek.medium.com



97



How to Use Nginx as a Reverse Proxy

Build a Gate Keeper For Your Servers

ziga-petek.medium.com

How to Use Private/Public Key Authentication With SSH

Forget your passwords on purpose!

ziga-petek.medium.com

[Dnsmasq](#)[Nginx](#)[Dns Servers](#)[Reverse Proxy](#)[Home Network](#)

Enjoy the read? Reward the writer.^{Beta}

Your tip will go to Ziga Petek through a third-party platform of their choice, letting them know you appreciate their story.

Give a tip

Sign up for CrunchX

By CodeX

A weekly newsletter on what's going on around the tech and programming space [Take a look.](#)

Your email



Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

