

Geis & Bell 2
Chase Geis & Francis Bell
COSC 362
How We Setup a Home Lab DNS Server

1 What is DNS?

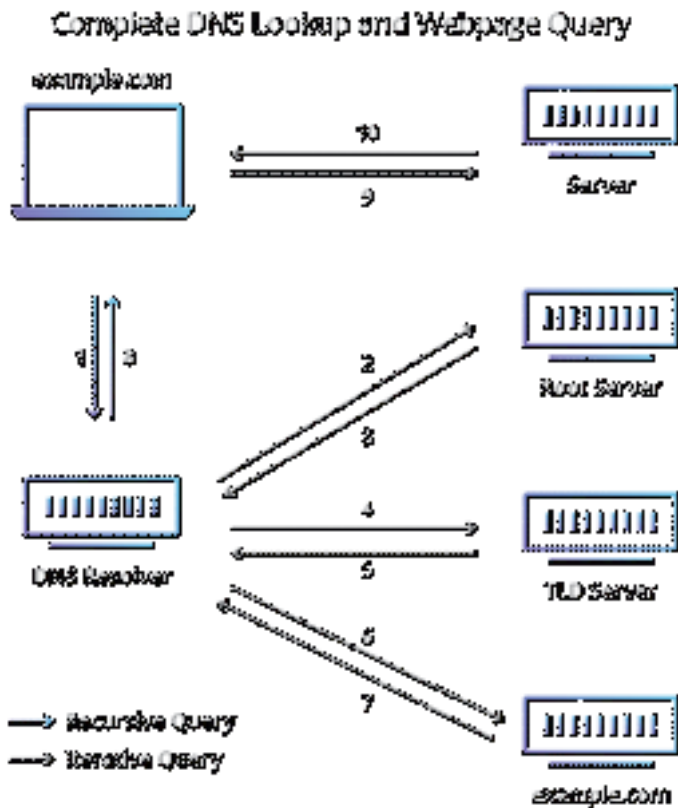
DNS Stands for a *Domain Name System*. It is the backbone of the internet that turns things that are easy for humans to remember and say like Google.com, Youtube.com, or LaTeXIsHardToUse.net into things that computers can read, IP addresses. If you ping Google.com through your command line you can get back the IP address. At the time of writing this the IP address of the Google.com server I was sent to was 216.58.192.206.

2 The Types of DNS Servers

There are different types of DNS servers though, some of them are private and can only be accessed and used from inside of a certain network, and some of them are public like your local ISP's (Internet Service Provider) DNS server. You can access these from anywhere. Ontop of the Public and Private DNS servers, there are also DNS servers that are called Recursive and Authoritative. Recursive DNS servers are your first stop for DNS traffic. They specialize in figuring out who to ask for the IP address of a website. It will recursively ask around to DNS servers and use the answers it gets back to find the all-knowing DNS server of the website you are looking for which is also know as an Authoritative DNS server. It is responsible for holding all the IP addresses and information for a specific website.

3 How DNS Works.

A good example of this is to think of a Google search. Google is like your recursive DNS server, you ask it a question, and it will try to find the answer for you and give it back to you (in the form of search results) and the website with the correct answer on it is the authoritative DNS server. Here is a nice example of a DNS query.



4 How We did DNS

There are many ways to do DNS. Every operating system has different DNS solutions but since we are Linux, we were stuck to use solutions native to that. There were 3 big options that we had for DNS solutions. The most comprehensive and all around feature full option in Bind. It is a full blown enterprise grade DNS solution that you can set up for free at home. The thing about it being enterprise though is that it is complex and has a LOT of options that you can use to configure the service. We actually had Bind fully running and working on our network before we decided not to proceed with it since it was overly complex for what we needed to do. It did gives us a good insight into what setting something like this up would be. We also could have gone with a solution called PowerDNS. It is very similar to Bind but it uses databases to get all of its information stored instead of using random config files around the system. This was even more complex than Bind for this reason so we decided to let it sit on the bench for now as well. We finally landed on a package called DNSMsd. It is a fully featured DNS and DHCP service that you can setup for your network. We decided just to use it for the DNS purposes though and did

not add any of the DHCP features. This gave us the simplicity we needed for our small network but still gave us all the great functionality of a DNS server.

5 How to Install and Configure DNSMasq

So how did we configure DNSMasq? Actually pretty simply. Lets walk through the steps to install and configure it for your home lab.

1. Open your command prompt and type the following command to update your repositories:

```
sudo apt-get update
```

1. Once that downloads, go ahead and install dnsmasq:

```
sudo apt install dnsmasq
```

1. Once you download dnsmasq, it will put all the default files where they need to be. You will need to change some settings in the default config file. Open it with the following command:

```
sudo nano /etc/dnsmasq.conf
```

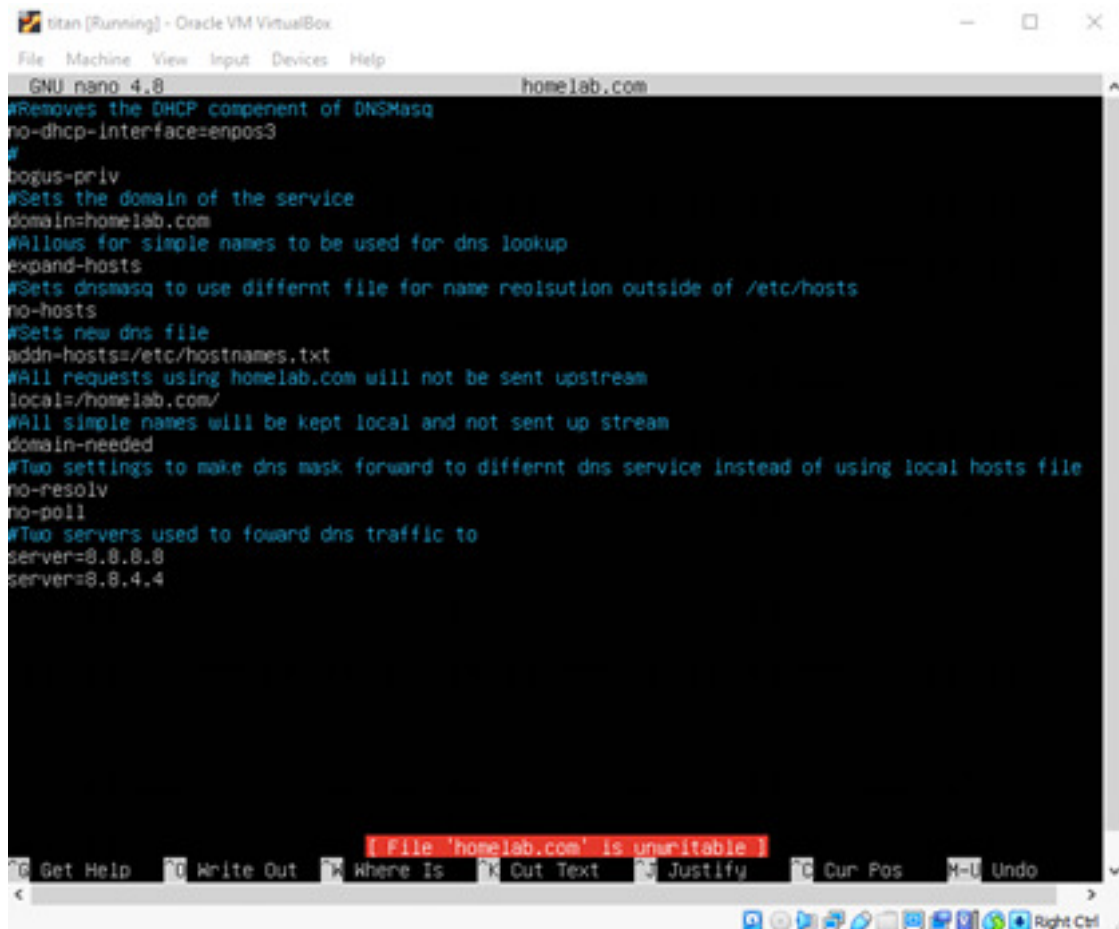
1. Find the line the the file that says “*#conf-dir=/etc/dnsmasq.d*” Remove the “*#*” from the lin to uncomment it from the file and go ahead and save the file using *control* + *X* and then type “*Y*”. This will save the file and then you can hit *control* + *X* again.

1. Once you exit the file we are ready to go and create a new config file for our homelab. We are going to use **Homelab.com** for our example so feel free to change that anywhere you see that to your own personal domain you want to configure. If you are in the /etc folder after exiting the file you can use the first command below. If you are in your home directory still you can use the second command:

```
sudo nano /dnsmasq.d/Homelab.com
```

```
sudo nano /etc/dnsmasq.d/Homelab.com
```

1. This is going to be the file that we do the most work in. We are going to add a decent chunk of things into the file to configure our DNS server just the way we want it. Take a look at the picture below to see all of the settings we chose as well as the comments above them that



```
GNU nano 4.8 homelab.com
#Removes the DHCP component of dnsmasq
no-dhcp-interface=enpos3
#
bogus-priv
#Sets the domain of the service
domain=homelab.com
#Allows for simple names to be used for dns lookup
expand-hosts
#Sets dnsmasq to use different file for name resolution outside of /etc/hosts
no-hosts
#Sets new dns file
addn-hosts=/etc/hostnames.txt
#All requests using homelab.com will not be sent upstream
local=/homelab.com/
#All simple names will be kept local and not sent up stream
domain-needed
#Two settings to make dns mask forward to different dns service instead of using local hosts file
no-resolv
no-poll
#Two servers used to forward dns traffic to
server=8.8.8.8
server=8.8.4.4

[ File 'homelab.com' is unwriteable ]
Get Help Write Out Where Is Cut Text Justify Cur Pos M-U Undo
```

1. Once you add these changes you can exit the same way we exited the file in *Step 4*
1. Now that we have those settings we can edit the last file which will have all of our hostnames with IP addresses associated with them in it. Go ahead and CD to get back to your home directory and type the following command in to edit the last file for our DNS service:

`sudo nano /etc/Hostnames.txt`

1. This will open a new text file. The format for this file is an IP address and a hostname on the same line. You don't need to add the .Homelab.com to the end of it though since we added the "*expanded-hosts*" option in the config file above. You can see an example below of what it should look like. Once you add all of the IP addresses you want to have in your DNS service you can close and exit out of the nano editor windows just like we did in *Step 4* and *Step 8*:

192.168.1.19 Falcon
192.168.1.20 Titan
192.168.1.21 Electron

1. You can now start the service on your linux device so it starts acting as a DNS server:

```
sudo Systemctl start dnsmasq
```

1. Now that you have all of your config files setup you can now set your DNS server's IP address as the main DNS service on all of your devices. This is done differently on Linux and Windows. This can be changed in the /etc/resolve.conf file on Linux. You can do this by typing the first two commands into your Linux command line and then adding the last three lines into the text editor:

```
sudo rm -f /etc/resolve.conf  
sudo nano /etc/resolve.conf  
nameserver *insert static IP of your DNS server*  
nameserver 8.8.8.8  
nameserver 8.8.4.4
```

1. Changing it on windows is a more involved process. Since this is a linux class you can google it for yourself to figure it out. You can test it by using the nslookup command below:

```
nslookup *name of a device you put into your Hostnames.txt file*
```

1. This should return you back with an IP address you put into your Hostnames.txt file. If you got the correct IP address back, then you successfully configured your own DNS server!

6 You Setup your DNS Server! Now What?

DNS servers can simplify some of your work around your lab. Instead of remembering long strings of numbers for all your machines you can just type in a short hostname that you setup. In enterprise environments they are extremely important for different internal services that are running but for a home lab they just simplify some projects. Hopefully your new system comes in handy!