# Overview

This lab will implement a simple domain name server between the end clients host and the top level domain (TLD) DNS server. In this lab the *hades* server DNS will serve as the TLD, and the student will implement their own sub domain server on their Debian VM. The system will then be tested by using another client. The sub domain DNS server will be used to resolve local domain names passed to it.

Before starting the installation be sure your Debian VM is “clean”, i.e. it starts with no problems. You will need Apache on it from the previous lab. After installing and configuring DNS you will be able to use the host + domain name to access the html files instead of typing the IP address.

Part of this lab has you change from the IP address being assigned by DHCP to using a static assigned address (but you will keep the address assigned originally by DHCP) The DHCP assigned address has a finite lifetime. The duration should last for the duration of the lab (that is the DHCP server within *hades* will not assign that IP address to another machine for a certain period of time), but if you stop the lab partway through and then resume it another day you should check to see if that address has been reassigned to another machine. If it has been, you must update the static address to a new address.

# Step 1: Install Bind

* apt-get install bind9

# Step 2: Network Configuration

DNS needs to run on a server that has a static address. Ensure your DNS server has a valid perhminate IP address.

## CLI

Stop the interface and edit the following files (or use the scripts from a previous lab):

* interfaces
	+ /etc/
	+ make sure the ip is static, note the address and mask
* resolv.conf
	+ The file is /etc/resolv.conf
	+ Edit it so the nameserver is your VM’s address and the domain and search entries are localdomain

# Step 3: Edit DNS configuration files

***Notes:***

* The bind configuration files are located in ***/etc/bind*** (contains some default files)
* The default zone files are located in ***/var/cache/bind/*** (initially empty)
* The name *my.net* is used as the default domain in the following examples; replace *my* with your own uid.

Open a Terminal and with root authority:

* Change to the **/etc/bind** directory
* Back up the **named.conf.local** file before editing:
	+ cp named.conf.local named.conf.local.backup
* Edit with the editor of your choice the **named.conf.local** file e.g.:
	+ vim.tiny /etc/bind/named.conf.local
* Use the following textbox as an example and enter information for two zones into the file:

//

// Do any local configuration here

//

// Consider adding the 1918 zones here, if they are not used in your

// organization

//include "/etc/bind/zones.rfc1918";

zone "my.net" {

 type master;

 file "/var/cache/bind/db.my.net";

};

zone "50.168.192.in-addr.arpa" {

 type master;

 file "/var/cache/bind/db.192";

};

* + For the first zone:
		- The file name can be anthing you want in any directory you want
		- Convention:
			* located in /var/cache/bind/
			* Named db.your.domainname
	+ For the second zone:
		- Again, the directory and name can be anything you want
		- Convention:
			* Located in /var/cache/bind/
			* Named db + reverse order for the top 3 IP addresses
				+ eg db.50.168.192 for the 192,168.50,0 network
* Save and quit from your editor.

# Step 4: Create and Edit Zone Files

Open a Terminal, change to “root”

* Create the zone file for the first zone
	+ vim.tiny /var/cache/bind/db.my.uncc
	+ **Note**: the file name must match the file name in the above *named.conf.local* file, change the name in the above command to create the proper named file
* Modify the file to match the following text box:

$TTL 1d

my.net. IN SOA ns1.my.net. root.my.net. (

 2007102701 ;

 3h ;

 15 ;

 1w ;

 3h ;

)

my.net. IN NS ns1.my.net.

ns1.my.net. IN A 192.168.50.128

[www.my.net](http://www.jxia3.uncc). IN A 192.168.50.128

debian.my.net. IN A 192.168.50.128

kerberos.my.net. IN CNAME debian.my.net.

;

;Sub-domain

;

[www.us.my.net](http://www.us.jxia3.uncc). IN A 192.168.50.128

ftp.us.my.net. IN CNAME www.us.my.net.

* + Be sure to change the names in the file to match your UID names
	+ **Warning:** Be sure to put the ending “.” on the names!
* Create the zone file for the second zone
	+ vim.tiny /var/cache/bind/db.192
	+ Again match the name of this file with the name of the second zone reference in the **named.conf.local** file
* Edit this file to match the data shown in following text box:

$TTL 1d

50.168.192.in-addr.arpa. IN SOA ns1.my.net. root.my.net. (

 2007102701;

 3h;

 15;

 1w;

 3h;

)

50.168.192.in-addr.arpa. IN NS ns1.my.net.

128.50.168.192.in-addr.arpa. IN PTR www.my.net.

* + Again change the names to match your id:
		- *my* to your UID
		- The reverse addresses to match your reverse addresses
* Save and quit vim
* Reload bind with the following command:
	+ /etc/init.d/bind9 reload

# Step 5: Test DNS

Use **nslookup** to test DNS

* Open Terminal, change to “root”
* Type the following command:
	+ nslookup domain-name or IP-address
* You should get a result similar to the following:

debian-my:/# nslookup 192.168.50.128

Server: 192.168.50.128

Address: 192.168.50.128#53

128.50.168.192.in-addr.arpa name = www.my.net.

debian-my:/# nslookup www.my.net

Server: 192.168.50.128

Address: 192.168.50.128#53

Name: www.my.net

Address: 192.168.50.128

debian-my:/# nslookup ftp.us.my.net

Server: 192.168.50.128

Address: 192.168.50.128#53

ftp.us.my.net canonical name = www.us.my.net.

Name: www.us.my.net

Address: 192.168.50.128

## More than likely at this point nslookup will have failed due to typos in the previous actions. There may be a misspelling in the contents of a file, in the name of a file, or in the changing of a configuration file. The rest of the lab will be spent tracking down those errors. Use the hints in the troubleshooting section and the end of this lab to help you debug the DNS.

Once DNS is working try bringing up the web page using the domain name instead of the IP address. Document the web page can be accessed via the domain name (screenshot).

* Start a browser
	+ Check you can browse your Apache web site with the IP address
	+ Now use the URL www.UID.uncc/home.html to browse
	+ Next try the URL for the sub domain: www.us.UID.uncc/home.html

# Trouble Shooting

To debug a lot of information to help will be in the log files for the DNS. The log files can get rather large over time, and most likely the errors will have occurred recently (at the end of the file).

To look at the last 20 entries in the log files use the following commands:

**# tail -n 20 /var/log/syslog
# tail -n 20 /var/log/messages**

You may want to also try tail –f filename and see what it does.

There are also some command-line tools to help you debug files:

**# /usr/sbin/named-checkconf**
# /usr/sbin/named-checkzone path/filename

You can look up what these tools do yourself.

### Typical Typos:

For the TTL the first character is “$”, the number “1” preceeds the d, not charater “l” (low case of L).

The correct first line is: $TTL 1d not STTL ld (note the *$* AND # *1*)

## Backup the configuration files before do any changes!

Be careful about editing the configuration files. Linux is case sensitive.

***Comment out unused lines***

Put some thought into whether a line should be deleted or commented out. If it is commented out it is easy to restore later.

## Run your VM from /scratch directory

Bad things will happen if you don’t

## Before Step5, test the following:

1. Reload bind with the following command:
2. **#/etc/init.d/bind9 reload**
3. then check the syslog file for errors. Use the timestamp to ensure currency.
4. Check to see if your VM is set correctly DNS with the following command:
5. **cat /etc/resolv.conf**

If the DNS IP is 172.16.1.1, edit the file to use your VM’s IP address.

To make network changes it is safest and least frustrating to edit the interfaces and resolv.conf files and then reboot the VM. Using the GUI interface sometimes works, sometimes don’t.