12.5.2 Configure DNS Settings

Click one of the buttons to take you to that part of the video.

Configure DNS Setttings 0:00-0:08

In this demonstration we're going to look at how you configure name resolution settings on a Linux system.

---

hosts File 0:09-3:02

Let's talk about the hosts file first. I'm currently on a Fedora system and if I go into my '/etc' directory and if I run the 'ls' command, we should see that there is a file here named hosts.

Understand that your Linux file system is configured by default to use the hostname IP address mappings contained within this file to resolve hostnames into IP addresses. Let's go ahead and open up this file in a text editor, the vi text editor, press the Insert key.

Notice that the syntax used in this file is relatively simple, we have the IP address over here then we have one or more hostnames. This one has two hostnames assigned fs5.corpnet.com, as well as localhost.localdomain, and then we also can assign a nickname to the IP address, so kind of a short cut.

In this example all we have is the IP address of the loopback adaptor, 127.0.0.1. The hostname of the system is fs5.corpnet.com; it also has a secondary hostname assigned to it of localhost.localdomain, and it has a shortcut of localhost.

You're not stuck with whatever you have here; you can edit this file and make whatever changes you want.

For example, what I would like to do is associate this hostname here with the IP address assigned to my Ethernet adapter that's actually connected to the network segment, and then just use the 'localhost' hostname for the loopback adapter address.

The first thing I need to do is find out what my IP address on this system is. We'll open up a new terminal window and we'll run the 'ifconfig' command to see what that is. It is 10.0.0.160, so we'll come back over here and enter '10.0.0.160' and we will map that IP address to 'fs5.corpnet.com' and we will accordingly remove it from the loopback adapter.

Now 127.0.0.1 maps to localhost.localdomain and the localhost shortcut and we will map 10.0.0.160 to 'fs5.corpnet.com' with a shortcut of 'fs5'.

You can add additional entries to the hosts file as well.

For example, we could add an entry for some other system on the network. For example, I have this openSUSE Linux system also running on the same network segment as my Fedora system. If I use the 'ifconfig' command here, we see that it has an IP address of 10.0.0.228 and its hostname is 'openSUSE.corpnet.com.'

I can go back to the hosts file on my Fedora system and, first of all, enter in the IP address of that remote system, '10.0.0.228' and I can then add its hostname here which is 'openSUSE.corpnet.com' shortcut 'openSUSE'.

Let's press Escape and let's 'exit' the editor and write our changes to the file and use a quick 'cat' command to verify we made the changes correctly, looks good.

Test Configuration 3:03-4:13

Let's test the configuration. Let's begin by pinging ourselves: 'fs5'. This will basically send a ping request from my Ethernet adapter to my Ethernet adapter.

I'll type 'ping fs5'. You'll notice over here I used this shortcut for this hostname, so I don't actually have to put in the full name of 'corpnet.com.' When I do, notice that my system resolved the hostname fs5.corpnet.com to the IP address 10.0.0.160, which is what we put in our hosts file and then it sends the ping request to that IP address.

Okay, now let's do the same thing, but this time let's ping the openSUSE system over the network instead of just ourselves. 'ping openSUSE' and once again it consults the hosts file, and it resolves openSUSE.corpnet.com into its associated IP address of 10.0.0.228.

Let's go ahead and break out. Our hosts file is configured and it is working, and we're able to resolve hostnames and IP addresses using the entries that we put in it.

Problem with Using the hosts File 4:14-4:28

There is actually a big problem with using the hosts file for name resolution. Key among these is the fact that it's not very dynamic and it's not very portable. In this configuration we have a DNS server, and this is where all of our network hosts will send all of their name resolution requests.

---

resolv.conf File 4:29-8:07

For this to work we have to configure this Linux system with the IP address of the name server that we want it to send name resolution request to. This can be done in two different ways.

One way is to edit the /etc/resolv.conf file. This is the traditional option for setting up your name server. The syntax in this file is fairly simple. We press the Insert key down here, we'll come down and add a new line and then we simply enter 'nameserver' followed by the IP address of that DNS server that we want to use.

For this system I'm going to use a public DNS server that's on the internet that everyone can use as IP address '8.8.8.8', probably one of the most frequently used DNS servers in the world. With this configuration, whenever a user enters a domain name that the system needs to resolve into an IP address, it will contact this DNS server at this address, submit the domain name to it and the DNS server will return that host's associated IP address.

Also notice that this file includes a search option. This is a very useful thing because it specifies the domain name that will be used to complete an incomplete hostname being submitted. Notice in this case it is corpnet.com.

For example, if I were to open up a shell prompt and enter ping fs1 and I don't provide a full domain name--just fs1, which is the hostname of the system--then this parameter up here search says that whenever someone enters a hostname and they don't provide a full domain name, just go ahead and tack corpnet.com onto the end of it.

Therefore, if I were to type ping fs1, the hostname would be automatically converted into a fully qualified domain of fs1.corpnet.com. If, on the other hand, I were to enter a fully qualified domain name, such as 'fs5.mydomain.com', then this parameter would not be used, because I provided a fully qualified domain name; it doesn't need to add this onto the end.

I'm actually not going to save this, because I want you to notice that there is a second option for configuring your name server address, and that is to go into your configuration files in /etc/sysconfig/networkscripts and add these parameters to the appropriate interface configuration file. Let's go ahead and do that. Go over here to this other window we have opened up. We'll switch to our root user account and we'll go into '/etc/sysconfig/network-scripts' directory. Notice there is a configuration file in this directory named 'ifcfg-ens192' that is used to configure the network interface in this system.

Let's open it in the vi editor and we'll use the syntax that we see listed right over here. Scroll down Insert key, go to the end add a new line, and we'll enter 'DNS1=' and then the IP address of the DNS server that we want to use--in this case '8.8.8.8'.

Notice that we can enter additional DNS servers too in case this one isn't available. You can actually enter up to three--that's your primary, your secondary, and your tertiary DNS servers--and the idea behind doing this is that if for some reason this DNS server is down, you don't want all network communication to be cut off, so we have a second DNS server that has the same information as this one that we can use until this one comes back online.

We enter 'DNS2= 8.8.4.4'. Notice also that we can enter in the search domain this has the same functionality as this option up here in the resolv.conf file. We enter 'DOMAIN=corpnet.com' press Escape.

Application of Changes 8:08-8:50

In order to apply that change, we have to do an 'ifdown ens192' and then bring the interface back up with the ifup command, its configuration file will be reread, and the new parameters we put into it will be applied.

'ifup ens92'. Do an 'ifconfig' command. Our interface is back up and running, and you'll notice that the DNS server address is not specified here in the output of 'ifconfig' the way it is in Windows systems with 'ipconfig'.

Name Server Resolution Orders 8:51-9:52

Before we end this demo, there's actually one more file that we need to look at here and that is the 'cat /etc/nsswitch.conf' file. You can configure a lot of different things in the nsswitch file, but there's one in particular that we're concerned with today and that's this one right here--hosts.

This entry specifies the sequence in which name resolution should occur. Notice that we have files here and over here we have dns.

This is very important. What it's saying basically is that if a user requests a resource from a network host using a hostname, a domain name, the first place we're going to look to try to resolve that hostname into an IP address is the hosts file.

If it finds an entry in the hosts file, that's the entry that's going to be used. Only if it can't find an entry in the hosts file for the requested domain name will it go over here and send the request to our configured name server--our DNS server.

Security Exploits 9:53-11:07

This is a very important thing to remember, because most Linux distributions are configured in this way. Such that if there's a conflict--if we have an entry in the hosts file and the same entry on the DNS server and they resolve to different IP addresses--the one in the hosts file is going to win because it's going to be the one that is checked first.

It's not just Linux that does this; most other operating systems do exactly the same thing, including Windows. This is important to remember because there are security exploits floating around that take advantage of this fact.

These exploits add entries without your knowledge to your hosts file that will resolve commonly accessed domain names--hostnames and domain names--into rogue web servers. For example, it might add an entry to the hosts file for amazon.com. It will point not to the real amazon.com IP address but to a rogue web server somewhere else that has a web page set up on it that looks just like amazon but is not, which will then capture your username and password when you try to log in to make a purchase.

You just basically gave the attacker all of your authentication information and then they go and run up a very nice bill on the real amazon website, using your username and password.

Summary 11:08-11:18

That's how you configure name resolution on a Linux system. In this demonstration we first looked at this hosts file and then we looked at the resolv.conf file and then we ended this demonstration by talking about name server resolution orders in the nsswitch file.
