12.5.3 Testing Name Resolution

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

Name Resolution Testing 0:00-0:13

In this demonstration we're going to look at several utilities that you can use to test name resolution. We're going to look at nslookup, dig, host, and getent.

---

nslookup Utility 0:14-1:45

Let's begin by looking at the nslookup command. nslookup is actually an older utility and it's not even included by default in many distributions now, although you can install it manually with YUM or Zypper, depending on which distribution you're using.

It is installed on this openSUSE's distribution so we can just use it at the shell prompt. In order to use nslookup, you type 'nslookup' at the shell prompt like I've done here. Then you enter the hostname that you want your DNS server to resolve.

For example, let's do a lookup of 'www.Google.com'. When you do, the utility returns the IP address associated with the domain name that you specified. As you can see right here, it tells us that www.Google.com maps to an IP address of 216.58.192.4.

Also, notice up here that it lists the IP address of the DNS server to which it sent this name resolution request: 10.0.0.254. Notice down here it specifies that this is a non-authoritative answer. The reason that it's non-authoritative is because this server here, 10.0.0.254, does not host the google.com zone.

It has no record for www.google.com, it had to go up to the root level name servers, find out the IP address of a server that is authoritative for google.com, send the request to it, and then it pointed the nslookup command to that name server which is where it actually got the result right here.

Interactive/ Non-Interactive Modes 1:46-2:54

nslookup can be used in two different ways. It can be used in interactive mode and non-interactive mode. The example that we saw right here is an example of non-interactive mode. We simply ran the command and gave it a hostname; it went out and resolved the hostname into an IP address and exited out. Non-interactive mode works great if you need to resolve one single hostname.

However, if you need to resolve a lot of hostnames, you can run nslookup in interactive mode. To do this you just type 'nslookup' and then we can enter each domain name we want to resolve. One after another. When you're done, you type 'exit' and you go back to the shell prompt.

That's nslookup. nslookup works great; we've used it for many, many years. However, it does have one key weakness and that is the fact that it really doesn't display a lot of information. If you want more extensive information about the hostname that you're trying to resolve, then you use a different utility.

---

dig Utility 2:55-4:49

dig will display extensive information about the domain name that you're trying to resolve. It works in much the same manner as nslookup. We enter 'dig' at the shell prompt, followed by the hostname that we want to resolve--in this example, 'www.google.com'--and it will send a request to our DNS server for name resolution.

However, notice that the information it pulls from the DNS server is much more extensive. Up here in the question section, we have the hostname that we requested information about, and then in the answer section down here, it lists all of the IP addresses that are mapped to this hostname.

Down here, dig also displays the query time; in other words, how long it took to perform the name resolution. It also displays the IP address of the DNS server that was used to resolve that hostname.

If you don't specify a particular DNS server with the dig command, then the default DNS server that you've configured in your resolv.conf file is going to be used. However, you can also tell dig to send a request to a different DNS server.

The syntax is to enter 'dig' and then an '@' followed by the IP address or hostname of the DNS server that you want to use. For example, here we used '10.0.0.254'. This is an internal DNS server that's on my same local network segment that I sent me DNS request to.

I can bypass that DNS server and go straight out to a public DNS server out on the internet if I want to, such as '8.8.8.8', and then put in the hostname that we want to resolve--'www.google.com'. It goes out and performs the same look-up that it did before, but it goes through a different server to do so.

The information is the same but notice down here, that the server that performed the request is now different; it went to 8.8.8.8 instead 10.0.0.254.

host Command 4:50-6:21

In addition to dig, you can also use the host command to resolve hostnames. Where dig provides extensive name resolution information, host is more like nslookup in that it provides very simple, quick name resolution information. The syntax that you use with host is very similar to that which you use with nslookup.

You enter 'host', followed by the hostname that you want to resolve, such as 'www.Google.com.' When we run it, it goes out, performs a name resolution request, and it tells us what IP address is associated with the hostname that we specified.

Also, notice down here that it not only performed an IPv4 look-up, but it also performed an IPv6 lookup as well. Notice that dig and nslookup did not do that.

Just as with dig, if you don't specify a DNS server with the host command, it's going to use whatever DNS server you have configured in your /etc/resolv.conf file. If you want to use a specific DNS server, you enter the same command that you did before, but you tack on the IP address or hostname of the DNS server that you want to use at the end of the command, such as '8.8.8.8'.

This will bypass our 10.0.0.254 DNS server, go out on the internet to 8.8.8.8, and send it this name resolution request. Notice that the output is slightly different now. Now it tells us, "Hey, you're using that different DNS server this time and here's the information."

getent Command 6:22-8:12

In addition to host and dig, and nslookup, you can use the getent command to perform name resolution requests.

getent is special. Understand that nslookup, host, and dig each use a DNS server to resolve hostnames, but your Linux system doesn't actually work this way; Windows systems don't work this way either for that matter.

Instead, when these systems need to resolve a hostname to an IP address, they first consult their /etc/host file first. Only if the hostname being resolved is not found in the host file, will they consult the DNS server.

The getent command actually mimics the way your system really performs name resolution by querying your host file first. It will first check the host file and only if a record isn't found in host will it then query your DNS server.

The syntax is getent hosts, followed by the hostname that you want to resolve. For example, I actually have an entry in my /etc/host file for router.corpnet.com. I'm going to enter 'getent hosts router'; it looks in my host file first, it finds an entry there and then it stops.

It doesn't even bother querying my DNS server. It says, "Okay. I found an entry. router.corpnet.com maps to 10.0.0.254." On the other hand, if I use 'getent' to resolve a hostname that doesn't exist in my host file, 'www.testout.com' it will then go out to a DNS server and send it a name resolution request and then display the appropriate IP address that maps to that hostname that we specified.

Summary 8:13-8:24

Those are some of the tools that you can use to test name resolution in your network. We first looked at the nslookup utility, then we looked at the dig utility, and then we looked at the host command, and then we ended this demonstration by looking at the getent command.