12.1 Introduction

Having access to the network is a key feature of most Linux systems. Users want to surf the net, send and receive email and transfer files with other users.

Typically the programs that perform these functions (web browsers, email clients, etc.) are fairly easy to use. However, they all rely on an important feature: the ability of your computer to communicate with another computer. In order to have this communication, you need to know how to configure your system's network.

Linux provides you with several tools to both configure your network as well as monitor how it is performing. In this chapter you will learn how to use both GUI-based tools as well as command line tools.

12.2 Linux Essentials Exam Objectives

This chapter will cover the topics for the following Linux Essentials exam objectives:

Topic 4: The Linux Operating System (weight: 8)

- **4.4 Your Computer on the Network**
    - Weight: 2
    - Description: Querying vital networking settings and determining the basic requirements for a computer on a Local Area Network (LAN).
    - Key Knowledge Areas:
        - Internet, network, routers
        - Domain Name Service
        - Network configuration
    - The following is a partial list of the used files, terms, and utilities:
        - route
        - resolv.conf
        - IPv4, IPv6
        - ifconfig
        - netstat
        - ping
    - Things that are nice to know
        - ssh
        - dig

12.3 Basic Network Terminology

Before setting up a network or accessing an existing network, it is important to know some key terms that are related to networking. This section explores the terms you should be aware of. Some of the terms are basic and you may already be familiar with them, however others are more advanced.

Host: A *host* is basically a computer. However, many people have a more limited idea of what a computer is (like a desktop computer or a laptop). In reality, many other devices are also computers, such as cell phones, digital music players and many modern televisions. In networking terms, a host is any device that communicates with another device.

Network: A *network* is a collection of two or more hosts (computers) that are able to communicate with each other. This communication can be via a wired connection or wireless.

Internet: The *Internet* is an example of a network. It consists of a publically accessible network that connects millions of hosts throughout the world. Many people use the Internet to surf web pages and send/receive email, but the Internet has many additional capabilities besides these activities.

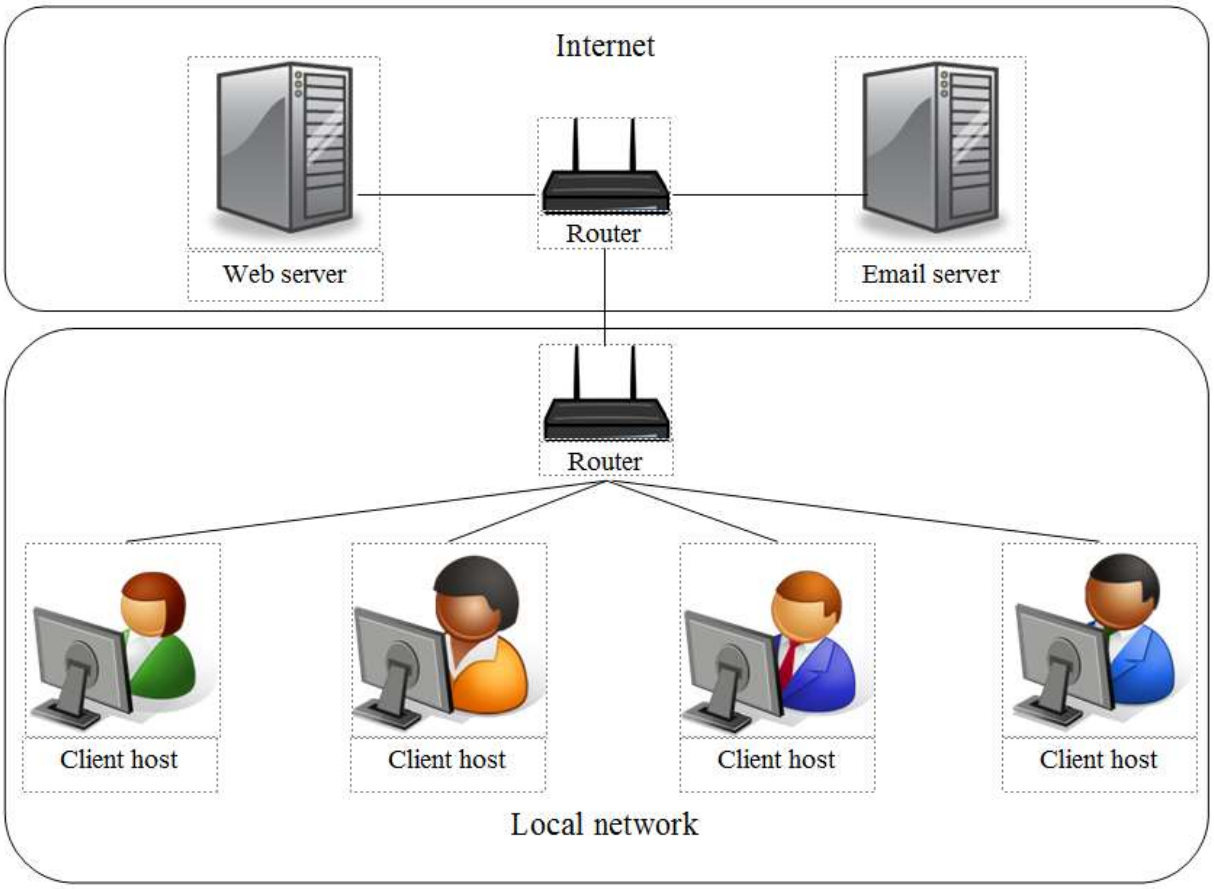
Wi-Fi: The term *Wi-Fi* refers to wireless networks.

Service: A host that provides some feature to another host is called a server. The feature being provided is the *service*. An example of a service would be when a host provides web pages to another host.

Client: A *client* is a host that is accessing a server. When you are working on a computer surfing the Internet, you are considered to be on a client host.

Router: Also called a *gateway*, a *router* is a machine that connects hosts from one network to another network. For example, if you work in an office environment, the computers within the company can all communicate via the *local network* created by the administrators. To access the Internet, the computers would have to communicate with a router that would be used to forward network communications to the Internet. Typically when you communicate on a large network (like the Internet), there are several routers that are used before your communication reaches its final destination.

The following diagram provides a visual picture of the terms discussed on the previous page:



12.4 Networking Features Terminology

In addition to the networking terms discussed in the last section, there are some additional terms that you should be aware of. These terms focus more on the different types of networking services that are commonly used as well as some of the techniques that are used to communicate between machines.

Network packet: A *network packet* is used to send network communication between hosts. By breaking down communication into smaller chunks (packets), the data delivery method is much more efficient.

IP address: An *Internet Protocol (IP) address* is a unique number assigned to a host on a network. Hosts use these numbers to "address" network communication. More discussion on IP addresses will occur later in this chapter.

Network mask: Also called a *netmask* or *mask*, a *network mask* is a number system that can be used to define which IP addresses are considered to be within a single network. Because of how routers perform their functions, networks have to be clearly defined.

Hostname: Each host on a network could have its own *hostname*. This makes it easier for humans to address network packets to another host because names are easier for humans to remember

than numbers. Hostnames are translated into IP addresses before the network packet is sent on the network.

DHCP: Hosts can be assigned hostnames, IP addresses and other network-related information by a *DHCP (Dynamic Host Configuration Protocol) server.* In the world of computers, a protocol is a well-defined set of rules. DHCP defines how network information is assigned to client hosts and the DHCP server is the machine that provides this information. While setting up a DHCP server is beyond the scope of this chapter, you will see how to configure a DHCP client machine later in this chapter.

DNS: As mentioned previously, hostnames are translated into IP addresses, prior to the network packet being sent on the network. This means that your host needs to know the IP address of all of the other hosts that you are communicating with. When working on a large network (like the Internet), this can pose a challenge as there are so many hosts. A *DNS (Domain Name Server) server* provides the service of translating IP addresses to hostnames. While setting up a DNS server is beyond the scope of this chapter, you will see how to configure a DNS client machine later in this chapter.

Ethernet: In a wired network environment, *Ethernet* is the most common way to physically connect the hosts into a network. Ethernet cables are connected to network cards that support Ethernet connections. Ethernet cables and devices (such as routers) are specifically designed to support different speeds of communications, the lowest being 10 Mbps (10 Megabits per second) and the highest being 100 Gbps (100 gigabits per second). The most common speeds are 100 Mbps and 1 Gbps.

TCP/IP: The *Transmission Control Protocol/Internet Protocol (TCP/IP)* is a fancy name for a collection of protocols (remember, protocol = set of rules) that are used to define how network communication should take place between hosts. While it isn't the only collection of protocols used to define network communication, it is the most often utilized one. As an example, TCP/IP includes the definition of how IP addresses and network masks work.

12.5 IP Addresses

As previously mentioned, hosts "address" network packets by using the IP address of the destination machine. The network packet also includes a "return address", the IP address of the sending machine.

There are, in fact, two different types of IP addresses: IPv4 and IPv6. To understand why there are two different types, you need to understand a brief bit of IP addressing history.

For many years, the IP addressing technique that was used by all computers was IPv4 (IP version 4). In an IPv4 address, a total of four 8-bit (8-bit = numbers from 0 to 255) numbers are used to define the address. For example: **192.168.10.120**. Note, this is considered a 32-bit address (4 x 8-bit = 32).

Each host on the Internet must have a unique IP address. In an IPv4 environment, there is a technical limit of about 4.3 billion IP addresses. However, many of these IP addresses are not really useable for various reasons. Also, IP addresses have been assigned to organizations that haven't fully make use of all of the IP addresses they had available.

While it seems like there should be plenty of IP addresses to go around, various factors (the increasing number of hosts on the Internet, reserved private IP addresses, etc.) led to a problem: The Internet started running out of IP addresses.

This, in part, encouraged the development of IPv6. IPv6 was officially "created" in 1998. In an IPv6 network the addresses are much larger, 128-bit addresses that look like this: **2001:0db8:85a3:0042:1000:8a2e:0370:7334**. Essentially this provides for a much larger address pool, so large that running out of addresses any time in the near future is very unlikely.

It is important to note the difference between IPv4 and IPv6 isn't just "more IP addresses". IPv6 has many other advanced features that address some of IPv4's limitations, including better speed, more advanced package management and more efficient data transportation.

Considering all the advantages, you would think that by now all hosts would be using IPv6. This isn't the case at all. The majority of network-attached devices in the world still use IPv4 (something like 98-99% of all devices). So, why hasn't the world embraced the superior technology of IPv6?

There are primarily two reasons:

- **The invention of NAT**: Invented to overcome the possibility of running out of IP addresses in an IPv4 environment, *Net Address Translation (NAT)* used a technique to provide more hosts access to the Internet. In a nutshell, a group of hosts are placed into a private network with no direct access to the Internet; a special router provides Internet access and only this one router needs an IP address to communicate on the Internet. In other words, a group of hosts share a single IP address, meaning a lot more computers can attach to the Internet. This feature means the need to move to IPv6 is less critical then before the invention of NAT.
- **Porting issues**: *Porting* is switching over from one technology to another. IPv6 has a lot of great new features, but all of the hosts need to be able to utilize these features. Getting everyone on the Internet (or even just some) to make these changes poses a challenge.

Most experts agree that IPv6 will eventually replace IPv4, so understanding the basics of both is important for those who work in the IT industry.

12.6 Configuring Network Devices

When you are configuring network devices, there are two initial questions that you need to ask:

- **Wired or wireless**? Configuring a wireless device will be slightly different than a wired device because of some of the additional features typically found on wireless devices (such as security).

- **DHCP or static address**? Recall that a DHCP server provides network information, such as your IP address and subnet mask. If you don't make use of a DHCP server, then you will need to manually provide this information to your host. This is called using a static IP address.

Generally speaking, a desktop machine will use wired network, while a laptop will use wireless. Normally a wired machine uses a static IP address, but these can also often be assigned via a DHCP server. In almost all cases, wireless machines use DHCP since they are almost always mobile and attached to different networks.

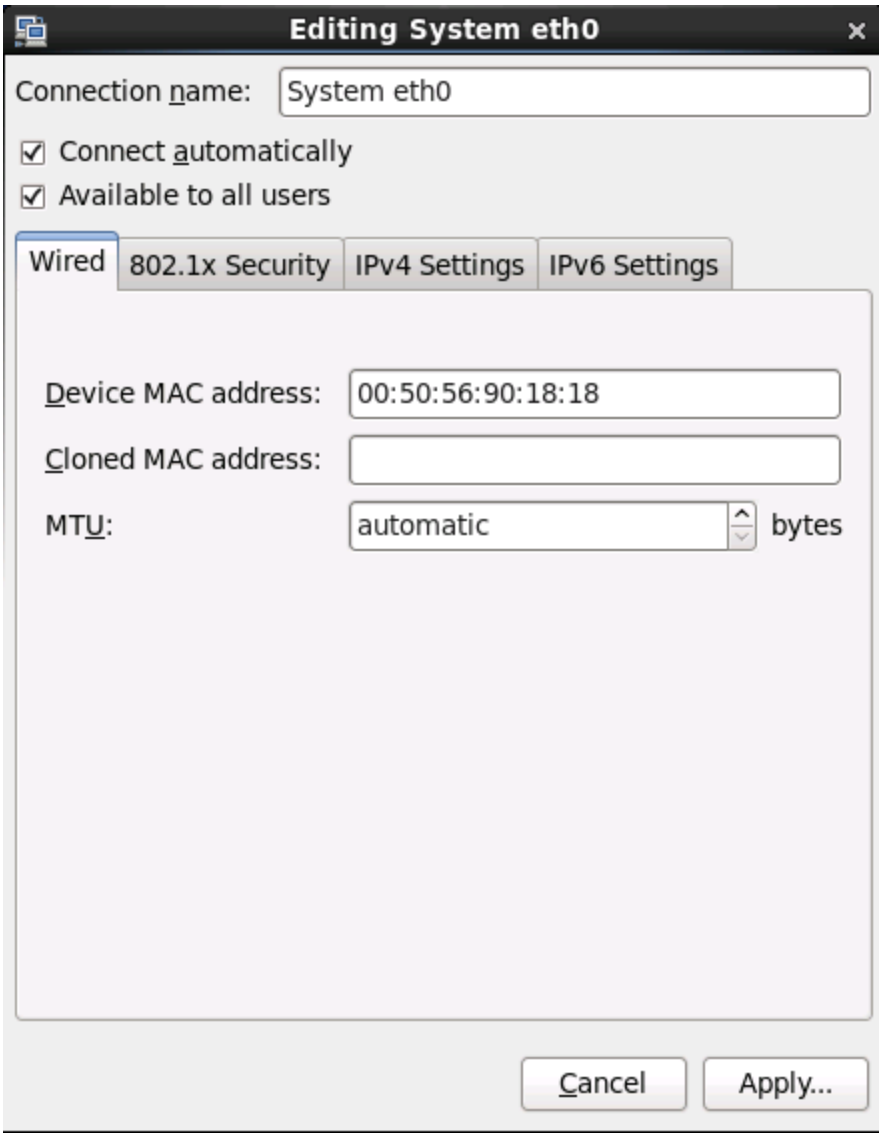
12.6.1 Configuring the Network Using a GUI

If you have access to a GUI (Graphical User Interface) environment, you will likely also have access to a GUI-based tool that will allow you to configure your network. These tools vary from one distribution to another. The following examples were performed on a CentOS machine.

To start the network configuration tool, click on **System** in the menu bar, then **Preferences** and then **Network Connections**:

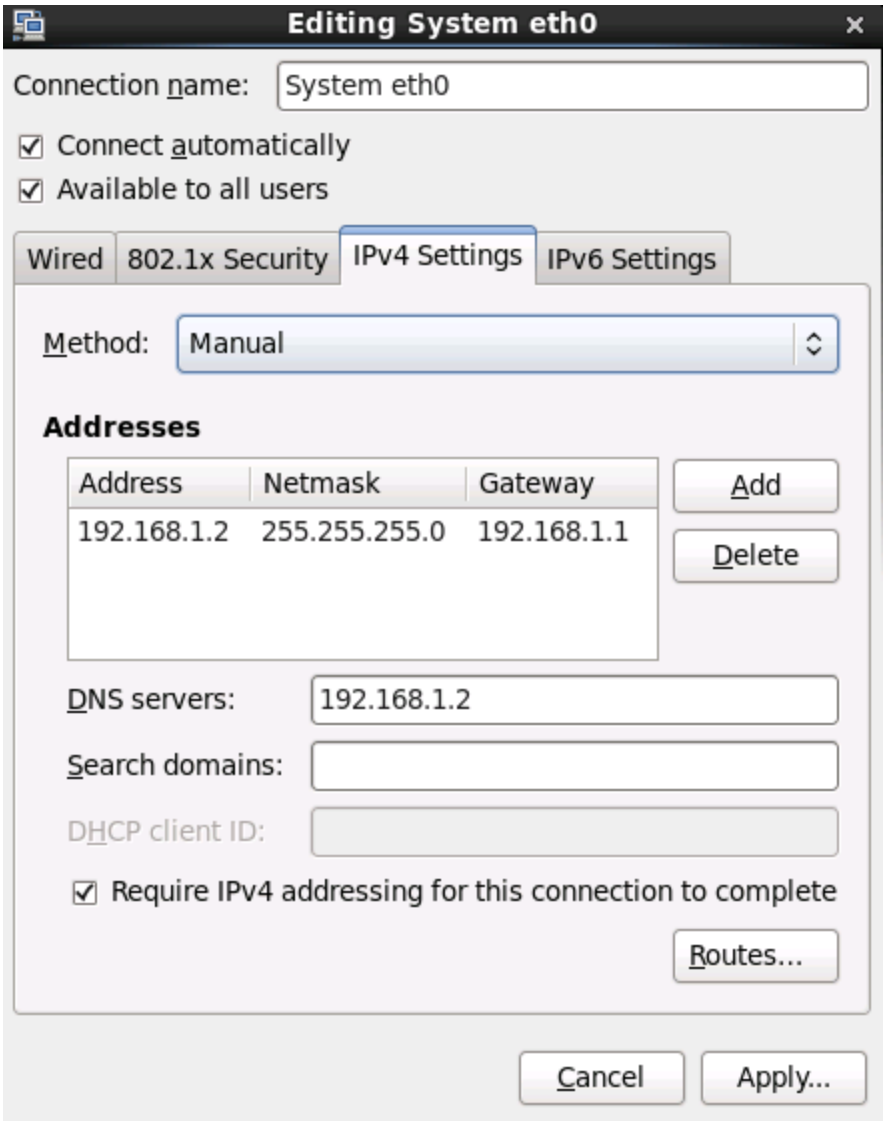
The tool first lists all of the current network devices. In the example below, there is only a **Wired** device:

The network device itself is **eth0**. Network devices are named **eth0**, **eth1**, etc. To modify this network device, click on the device name and then click the **Edit** button:

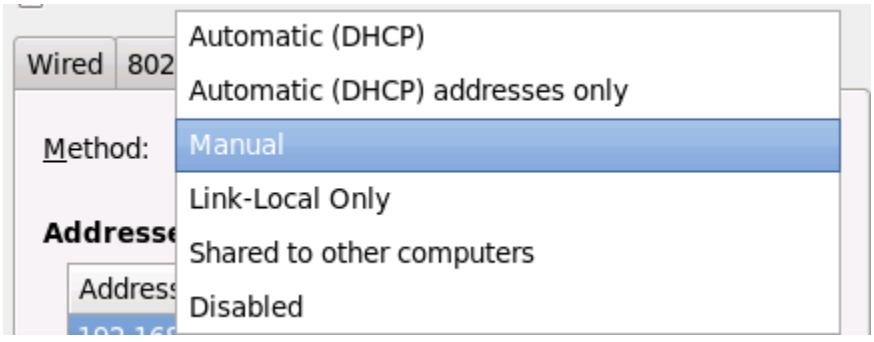


Note that a full discussion of all network features is beyond the scope of this course. The focus on this section will be to change key network components.

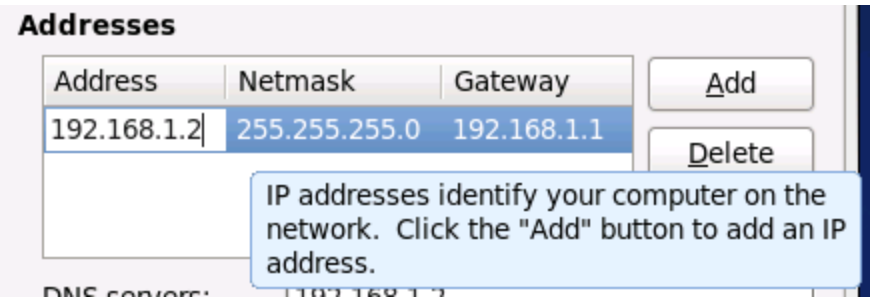
If you click on the **IPv4 Settings** tab, the following would be displayed:



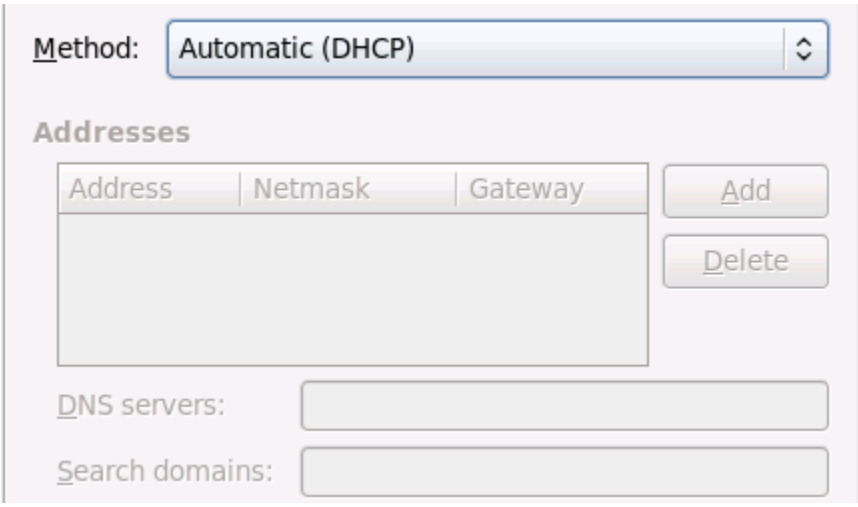
Recall that you can assign a static IP address or use a DHCP server (if one is available). This change can be made by clicking on the drop down list next to **Method**:



If you choose **Manual**, then you will be able to change the current address by clicking in the area where the address is currently specified:

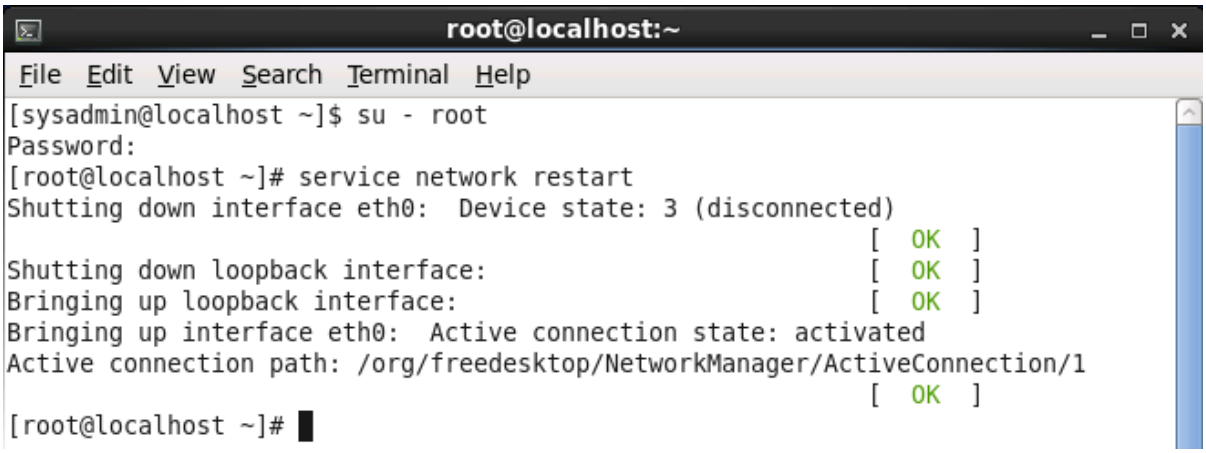


Note that if you choose **Automatic (DHCP)**, then the **Addresses** location is "grayed out":



If you switch from Automatic (DHCP) back to Manual, all of the previous data is "gone". By clicking the Cancel button and editing the eth0 device again, the data will reappear.

Most GUI-based tools make changes take effect immediately after you save them. However, in some cases, you might need to either reboot the machine or run a command as the administrator to make the changes take effect. The following demonstrates the command that would need to be executed on a CentOS system:



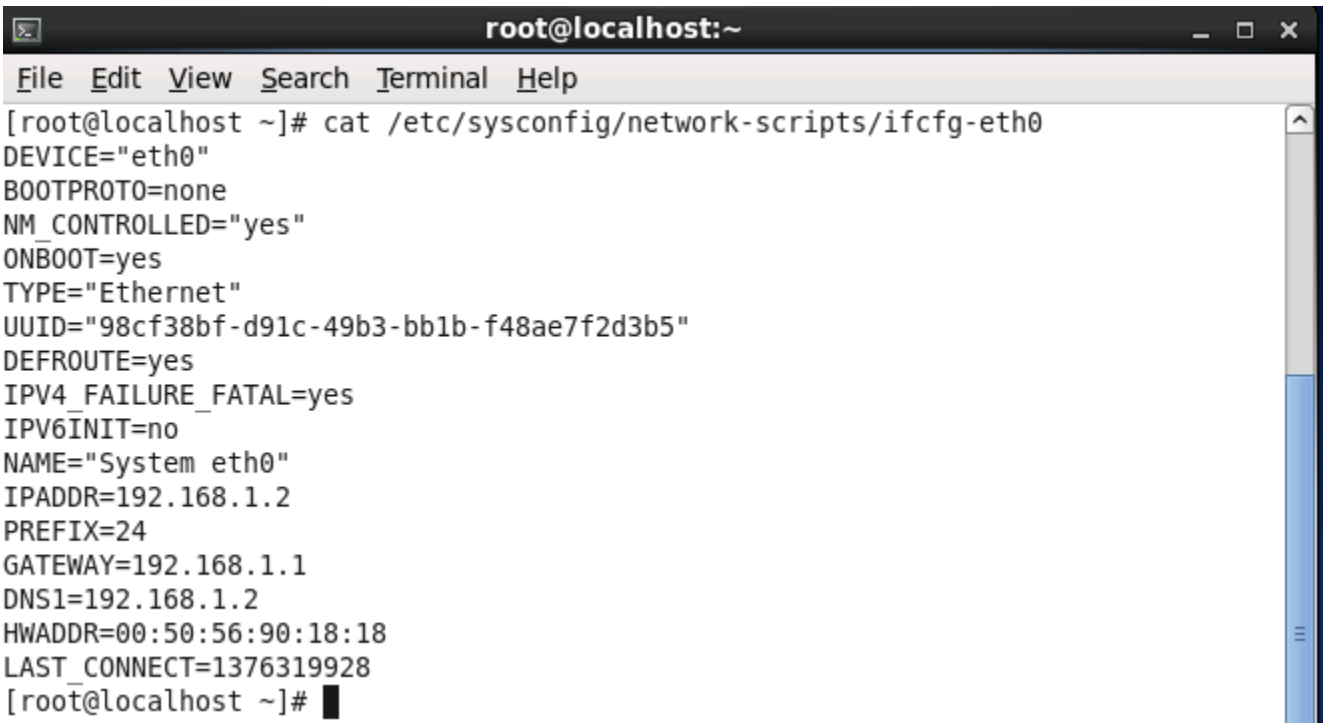
12.6.2 Configuring the Network Using Configuration Files

There will be times when no GUI-based tool will be available. In those cases, it is helpful to know the configuration files that are used to store and modify network data.

These files can vary depending on the distribution that you are working on. The following examples are provided for CENTOS systems.

12.6.2.1 Primary IPv4 Configuration File

The primary configuration file for an IPv4 network interface is the `/etc/sysconfig/network-scripts/ifcfg-eth0` file. The following demonstrates what this file looks like when configured for a static IP address:



If the device was configured to be a DHCP client, then the **IPADDR**, **GATEWAY** and **DNS1** values would not be set. Additionally, the **BOOTPROTO** value would be set to "dhcp".

12.6.2.2 Primary IPv6 Configuration File

On a CentOS system, the primary IPv6 configuration file is the same file where IPv4 configuration is stored: the `/etc/sysconfig/network-scripts/ifcfg-eth0` file. If you want to have your system have a static IPv6 address, add the following to the configuration file:

```
IPV6INIT=yes
```

```
IPV6ADDR=<IPv6 IP Address>
IPV6_DEFAULTGW=<IPv6 IP Gateway Address>
```

If you want your system to be a DHCP IPv6 client, then add the following setting:

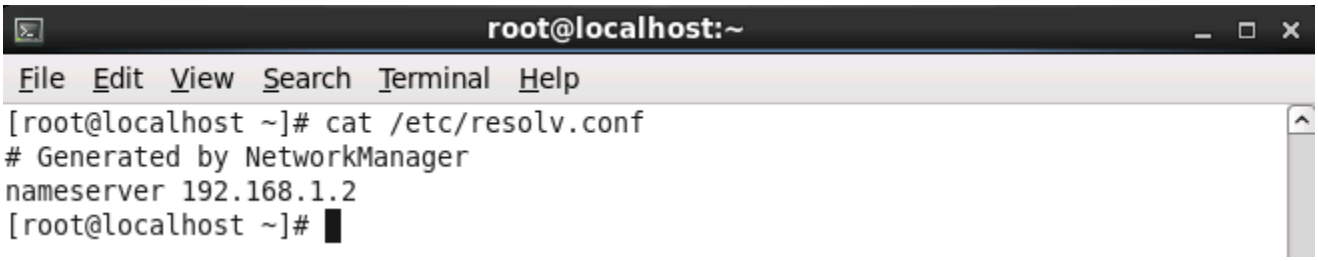
```
DHCPV6C=yes
```

You also need to add the following setting to the `/etc/sysconfig/network` file:

```
NETWORKING_IPV6=yes
```

12.6.2.3 /etc/resolv.conf File

The file that holds the location of the DNS server is the `/etc/resolv.conf` file. A typical `/etc/resolv.conf` file looks like the following:



The nameserver setting is set to the IP address of the DNS server. It is common to have multiple nameserver settings, in the event that one DNS server isn't responding.

12.6.2.4 Additional Network Configuration Files

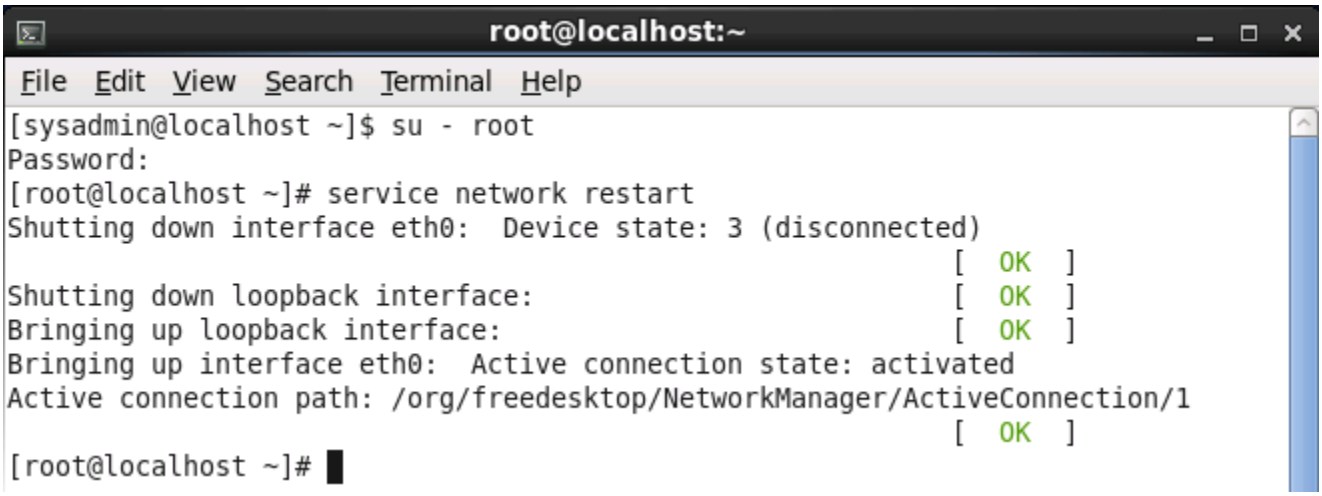
The following table describes additional network configuration files to be aware of. Although they are not specifically listed in the exam objectives, the objectives do include the general term "Network configuration", so these files may in fact appear on the exam:

Command	Explanation
/etc/hosts	This file contains a table of hostnames to IP addresses. It can be used to supplement a DNS server.
/etc/sysconfig/network	This file has two settings. The **NETWORK** setting can determine if networking is turned on (yes) or off (no). The **HOSTNAME** setting defines the local machine's hostname.

Command	Explanation
/etc/nsswitch.conf	This file can be used to modify where hostname lookups occur. For example, the setting **hosts: files dns** would have hostname lookups occur in the `/etc/hosts` file first and then the DNS server second. If switched to **hosts: dns files**, the DNS server would be searched first.

12.6.2.5 Restarting the Network

After changing a network configuration file (for example, the `/etc/sysconfig/network-scripts/ifcfg-eth0` file or the `/etc/resolv.conf` file), you either need to reboot the machine or run a command as the administrator to make the changes take effect. The following example demonstrates the command that would need to be executed on a CentOS system:

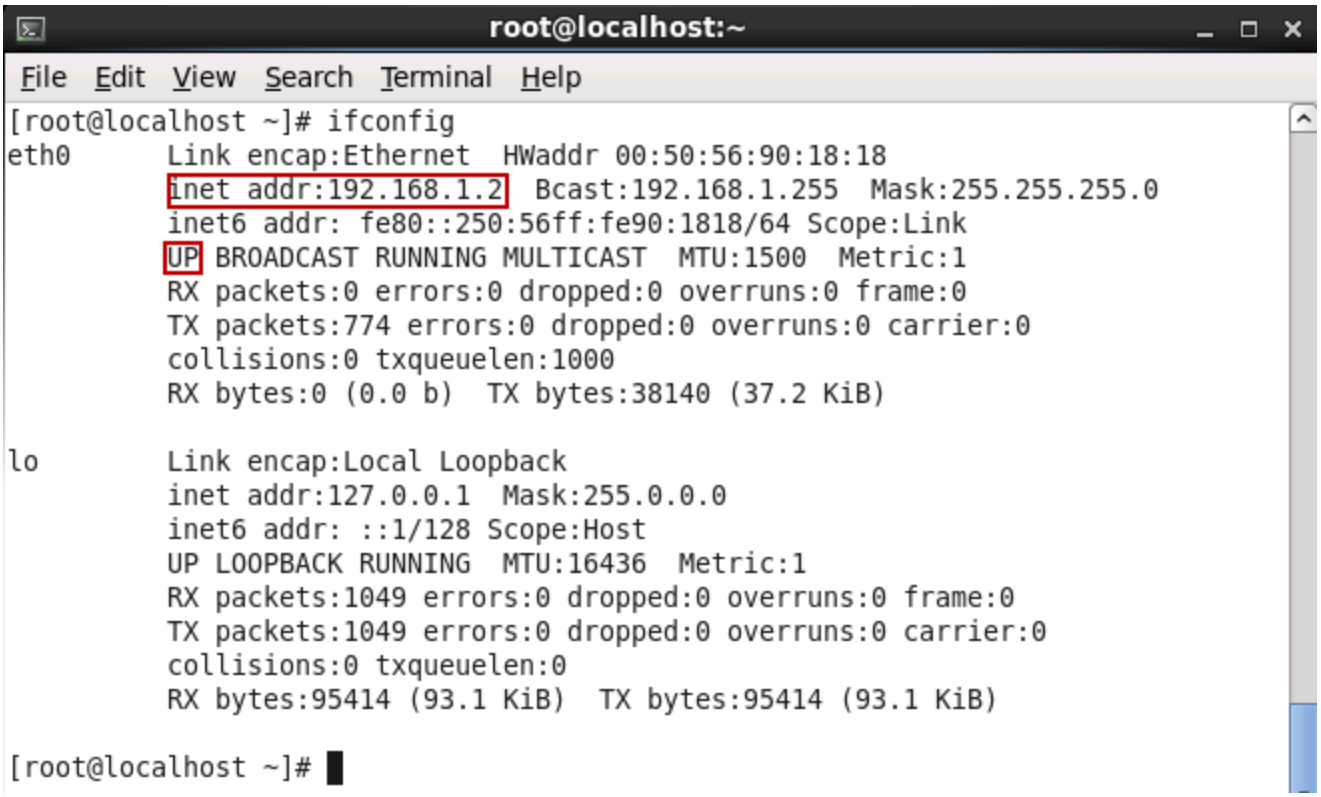


12.7 Network Tools

There are several commands that you can use to view network information. These tools can also be useful when you are troubleshooting network issues.

12.7.1 ifconfig Command

The `ifconfig` command is used to display your network configuration information. Again, not all network settings are covered in this course, but you can see from the output below that the IP address of the primary network device (eth0) is 192.168.1.2 and that the device is currently active (UP):

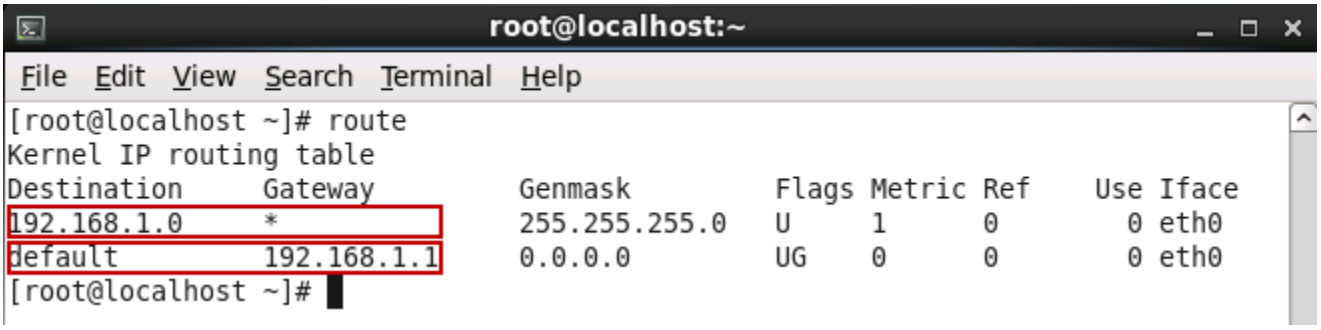


The **lo** device is referred to as the *loopback* device. It is a special network device used by the system when sending network-based data to itself.

The `ifconfig` command can also be used to temporarily modify network settings. Typically these changes should be permanent, so using the `ifconfig` command to make such changes is fairly rare.

12.7.2 route Command

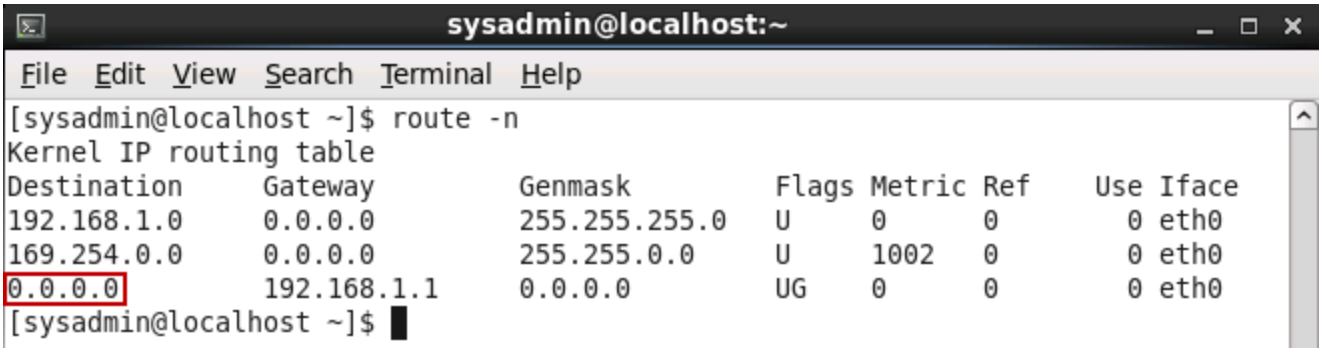
Recall that a router (or gateway) is a machine that will allow hosts from one network to communicate with another network. To view a table that describes where network packages are sent, use the `route` command:



The first red box in the example above indicates that any network package sent to a machine in the**192.168.1** network is not sent to a gateway machine (the * indicates "no gateway"). The second

red box indicates that all other network packets are sent to the host with the IP address of**192.168.1.1** (the router).

Some users prefer to display this information with numeric data only, by using the `-n` option to the`route` command. For example, look at the following and focus on where the output used to display **default**:



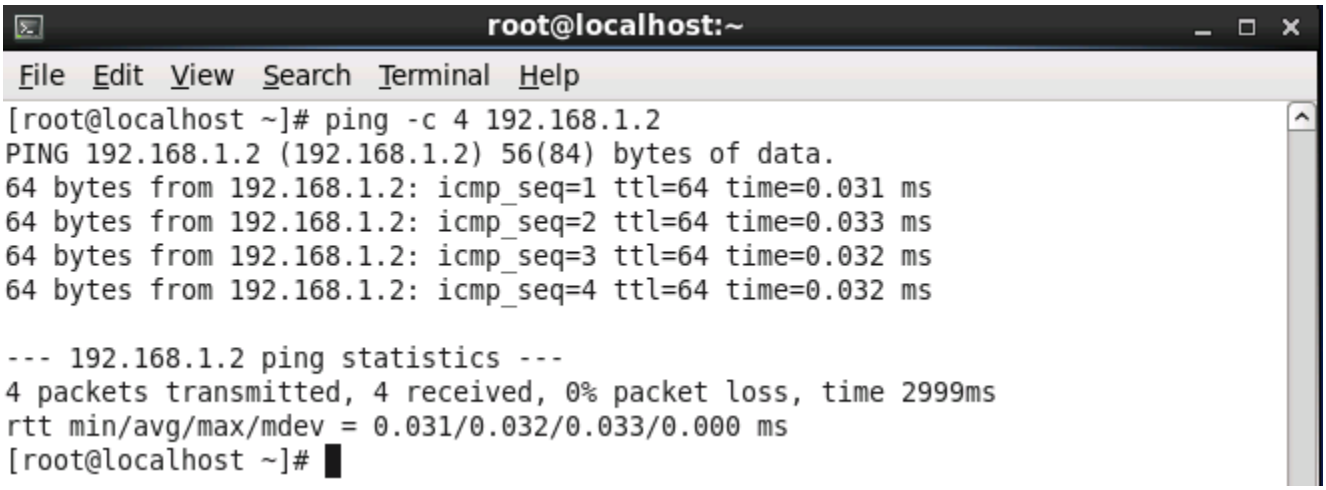
The 0.0.0.0 refers to "all other machines", or the same as "default".

12.7.3 ping Command

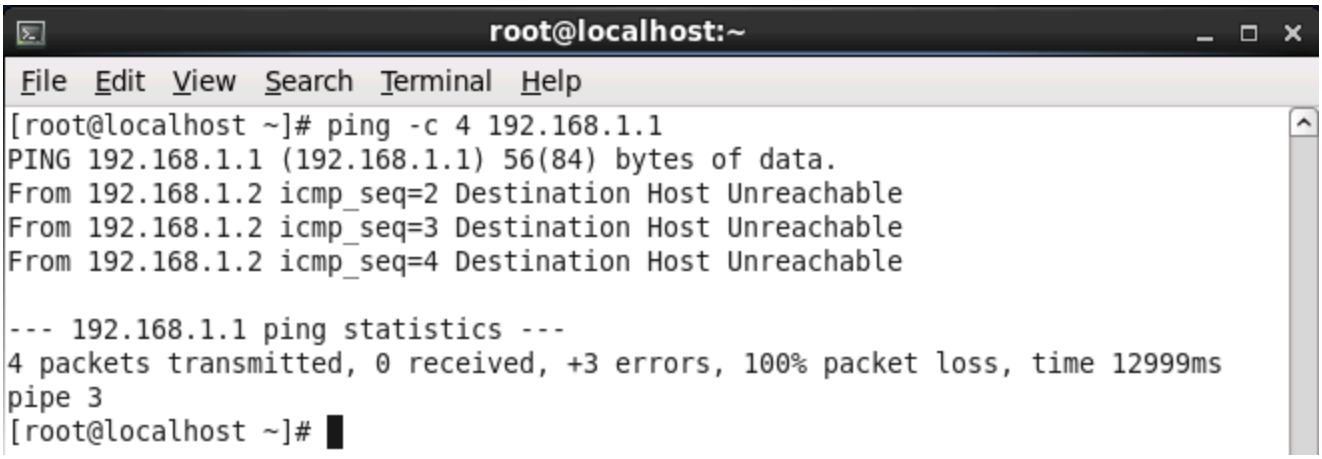
The `ping` command can be used to determine if another machine is "reachable". If the `ping` command can send a network package to another machine and receive a response, then you should be able to connect to that machine.

By default, the `ping` command will continue sending packages over and over. To limit how many pings to send, use the `-c` option.

If the `ping` command is successful, you will see output like the following:



If the `ping` command fails, you will receive a message stating, "Destination Host Unreachable":



It is important to note that just because the `ping` command fails does not mean that the remote system is really unreachable. Some administrators configure their machines to not respond to ping requests.

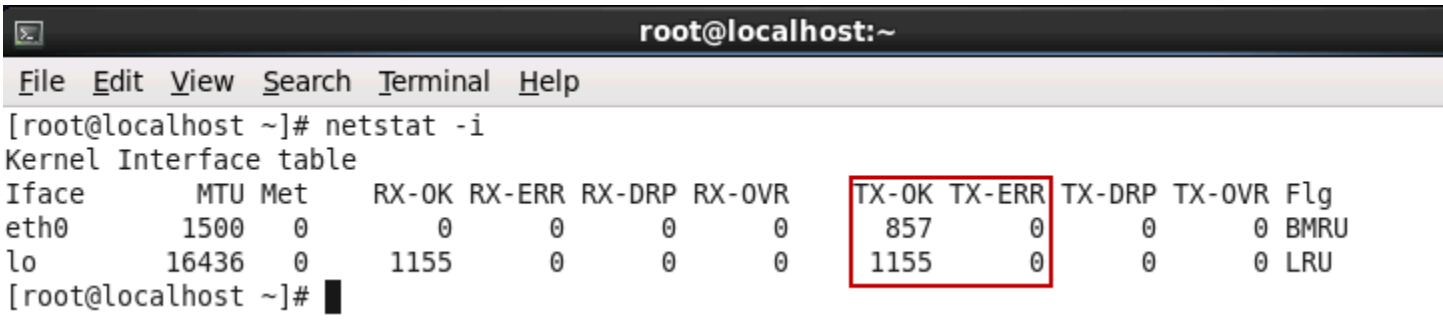
This is because a server can be attacked by something called a *denial of service attack*. In this sort of attack, a server is overwhelmed by a massive number of network packets. By ignoring ping requests, the server is less vulnerable.

As a result, the `ping` command may be useful for checking the availability of local machines, but not always for machines outside of your own network.

12.7.4 netstat Command

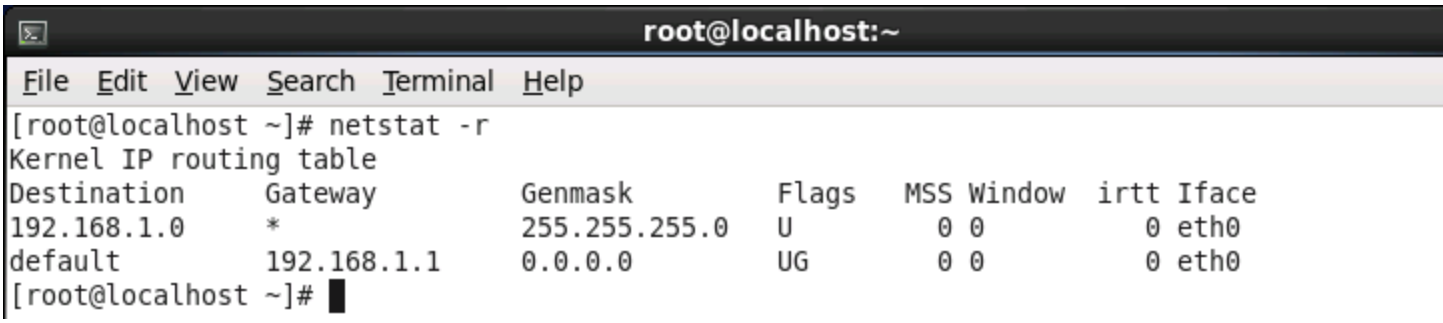
The `netstat` command is a powerful tool that provides a large amount of network information. It can be used to display information about network connections as well as display the routing table similar to the `route` command.

For example, you may want to display statistics regarding network traffic. This can be accomplished by using the `-i` option to the `netstat` command:



The most important statistics from the output above are the TX-OK and TX-ERR. A high percentage of TX-ERR may indicate a problem on the network, such as too much network traffic.

If you want to use the `netstat` command to display routing information, use the `-r` option:

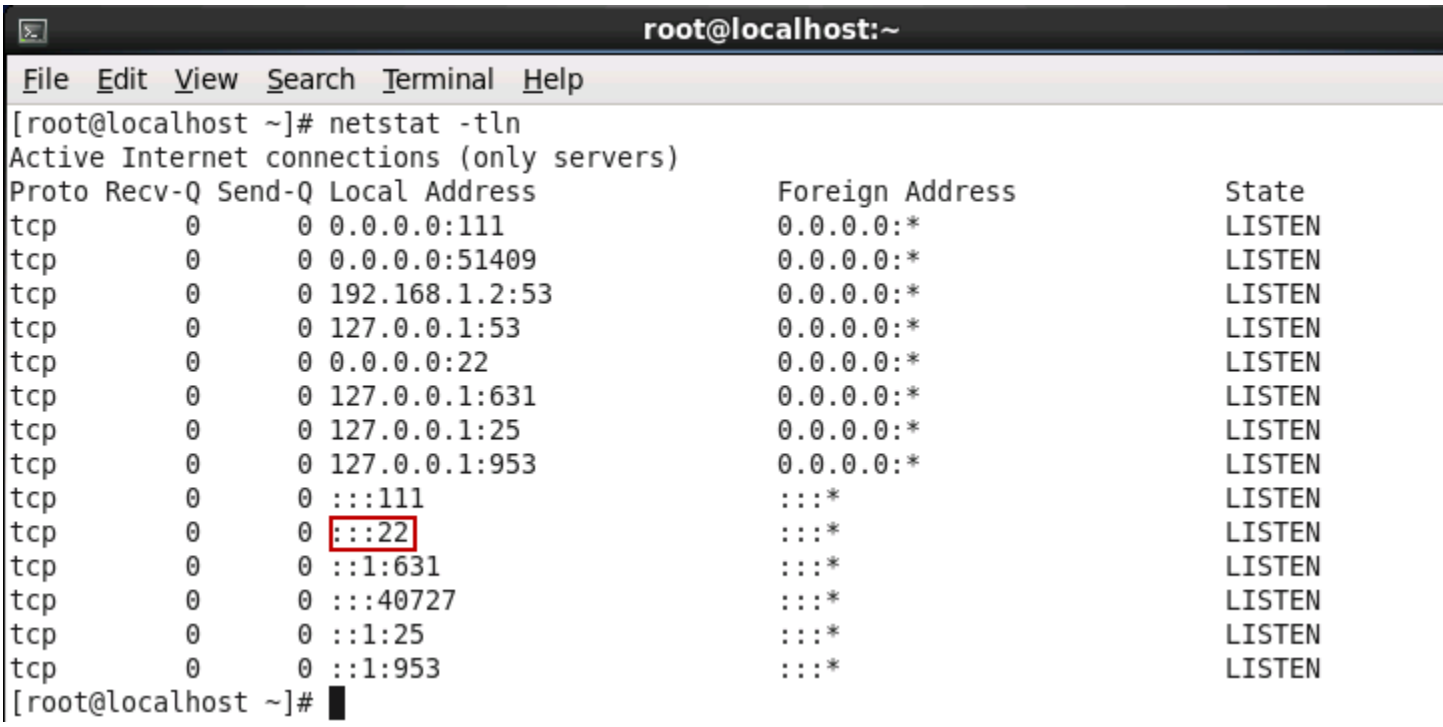


The `netstat` command is also commonly used to display open *ports*. A port is a unique number that is associated with a service provided by a host. If the port is open, then the service is available for other hosts.

For example, you can log into a host from another host using a service called *SSH*. The SSH service is assigned port #22. So, if port #22 is open, then the service is available to other hosts.

It is important to note that the host also needs to have the services itself running; this means that the program that allows remote users to log in needs to be started (which it typically is, for most Linux distributions).

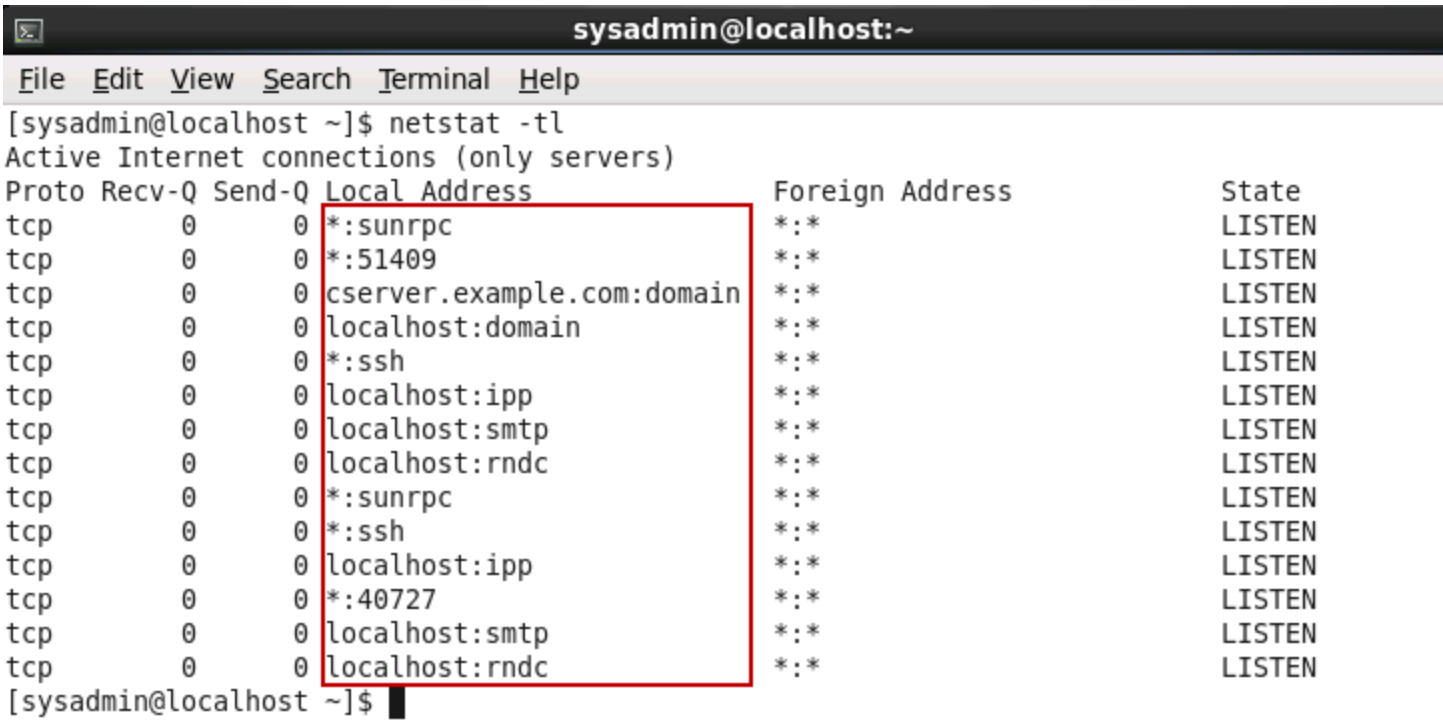
To see a list of all currently open ports, you can use the following command:



As you can see from the output above, port #22 is "LISTENing", which means it is open.

In the previous example, `-t` stands for TCP (recall this protocol from earlier in this chapter), `-l` stands for "listening" (which ports are listening) and `-n` stands for "show numbers, not names".

Sometimes showing the names can be more useful. Just drop the `-n` option:



On some distributions you may see the following message in the **man page** of the `netstat`command:

```
NOTE
       This program is obsolete.  Replacement for netstat is ss.  Replacement for netstat
       -r  is  ip route.  Replacement for netstat -i is ip -s link.  Replacement for net-
       stat -g is ip maddr.
```

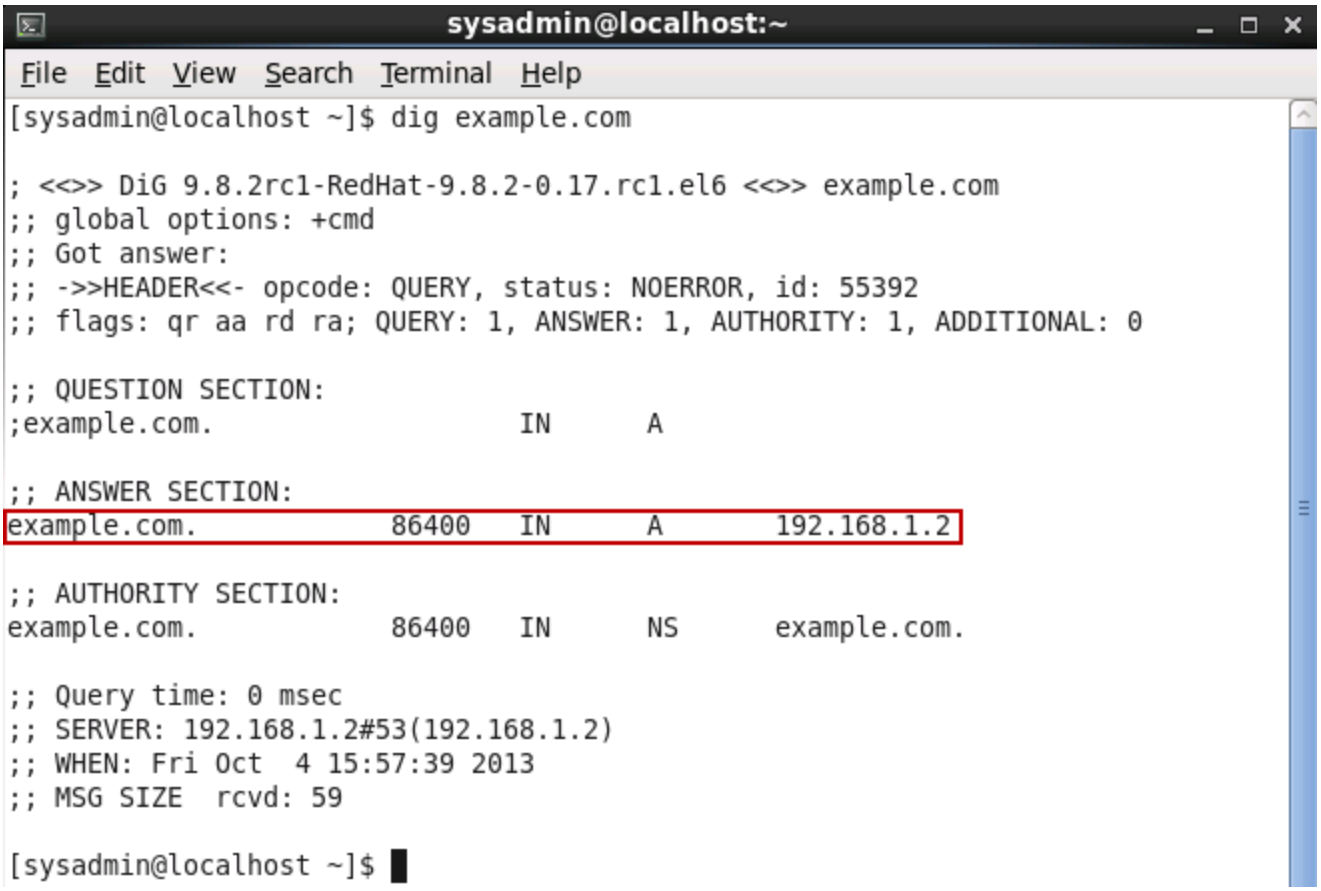
While no further development is being done on the `netstat` command, it is still an excellent tool for displaying network information. The goal is to eventually replace the `netstat` command with commands such as the `ss` and `ip` commands. However, it is important to realize that this may take some time.

The `netstat` command is covered in this course because it is available on all Linux distributions, still widely used and it is a Linux Essentials exam objective (the `ss` and `ip` commands are not).

12.7.5 dig Command

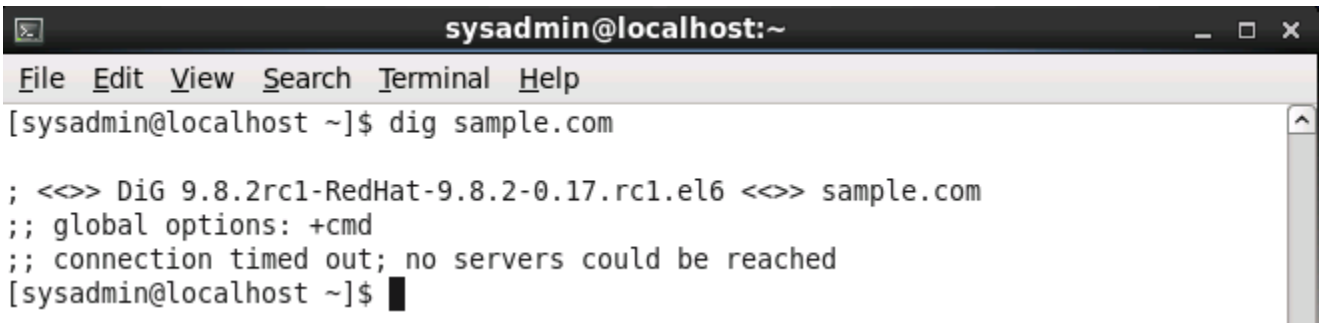
There may be times when you need to test the functionality of the DNS server that your host is using. One way of doing this is to use the `dig` command. This command will perform queries on the DNS server to determine if the information needed is available on the server.

In the following example, the `dig` command is used to determine the IP address of the example.com host:



Note that the response included the IP address of 192.168.1.2, meaning that DNS server has the IP address to hostname translation information in its database.

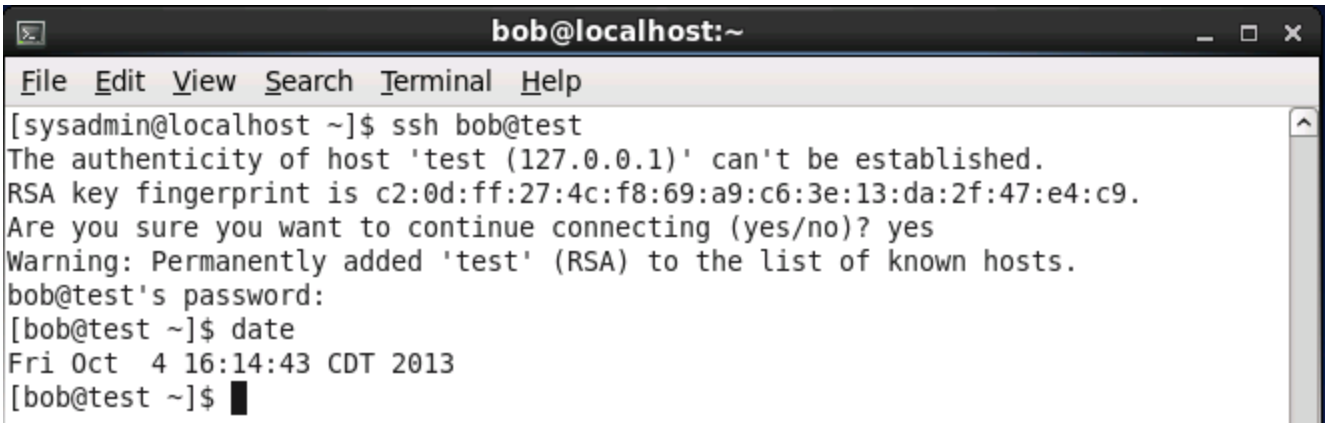
If your DNS server doesn't have the requested information, it is configured to ask other DNS servers. If none of them have the requested information, you will receive an error message:



12.7.6 ssh Command

The `ssh` command will allow you to connect to another machine across the network, log in and then perform tasks on the remote machine.

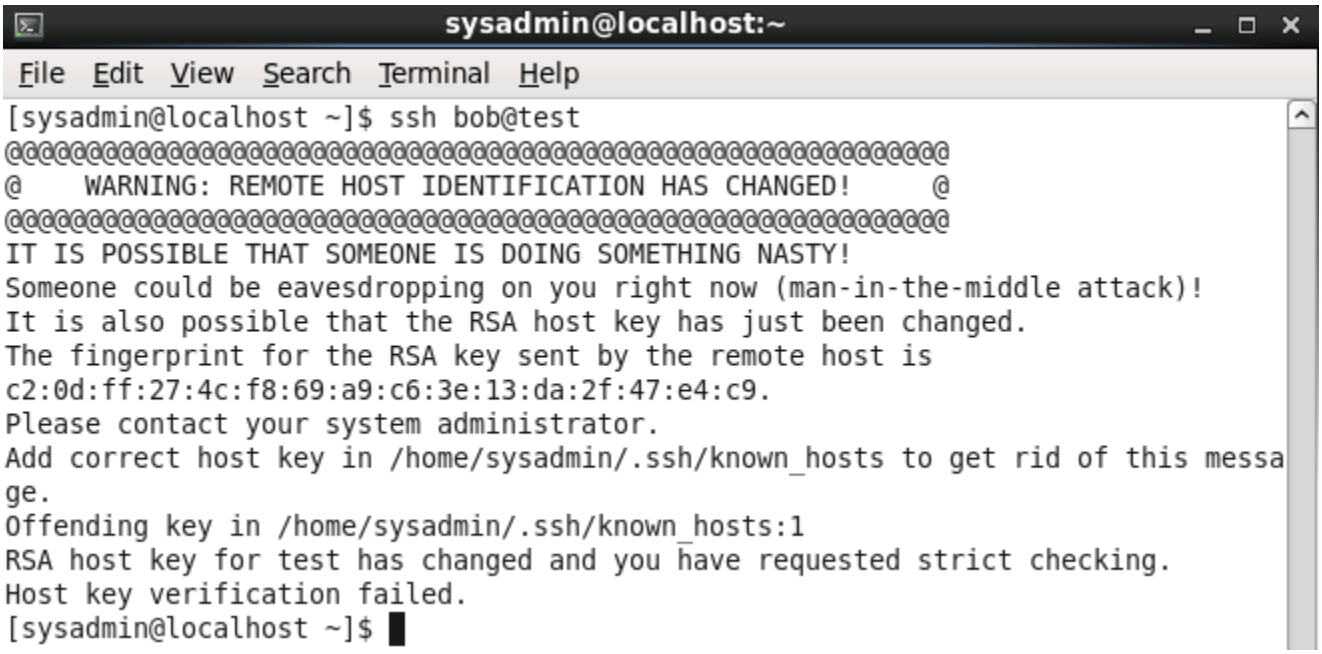
When you use the `ssh` command and only provide a machine name or IP address to log into, the command will assume you want to log in using the same username that you are currently logged in as. If you want to use a different username, use the syntax *username@hostname*:



12.7.6.1 RSA Key Fingerprint

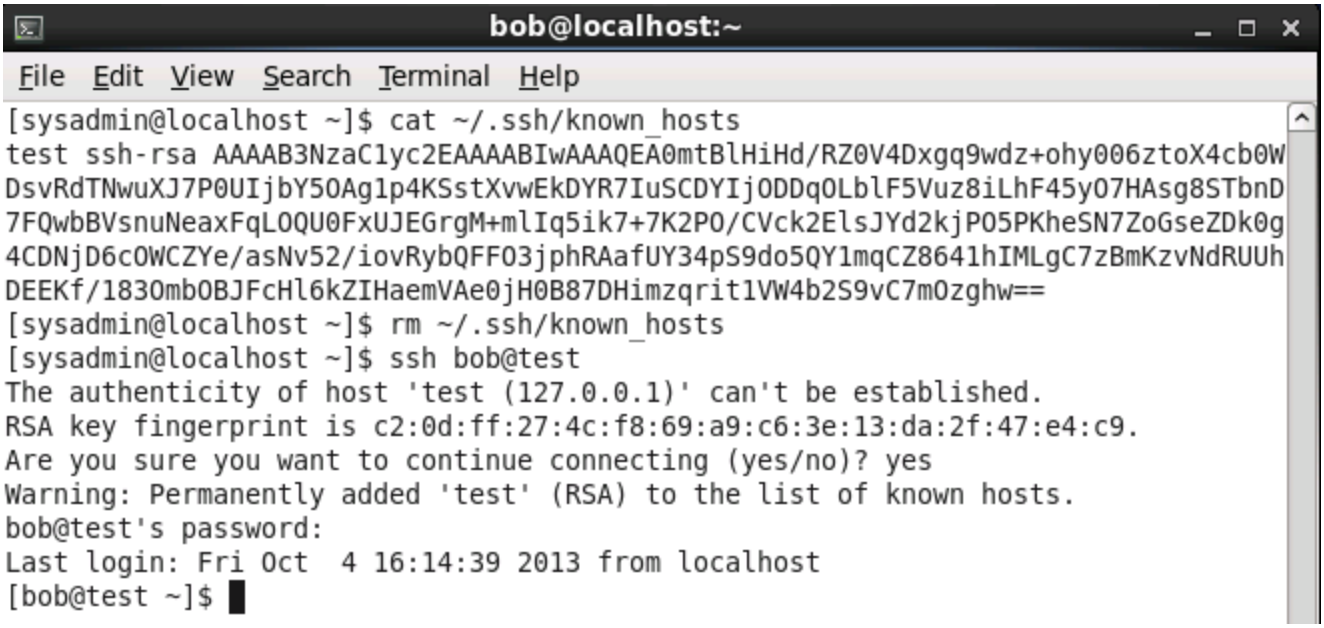
The first prompt asks you to verify the identity of the machine you are logging into. In most cases, you are going to want to answer "yes". While you can check with the administrator of the remote machine to make sure that the RSA key fingerprint is correct, this isn't really the purpose of this query. It is really designed for future log in attempts.

After you answer "yes", the RSA key fingerprint of the remote machine is stored on your local system. When you attempt to ssh to this same machine in the future, the RSA key fingerprint provided by the remote machine is compared to the copy stored on the local machine. If they match, then the username prompt appears. If they don't match, you will see an error like the following:



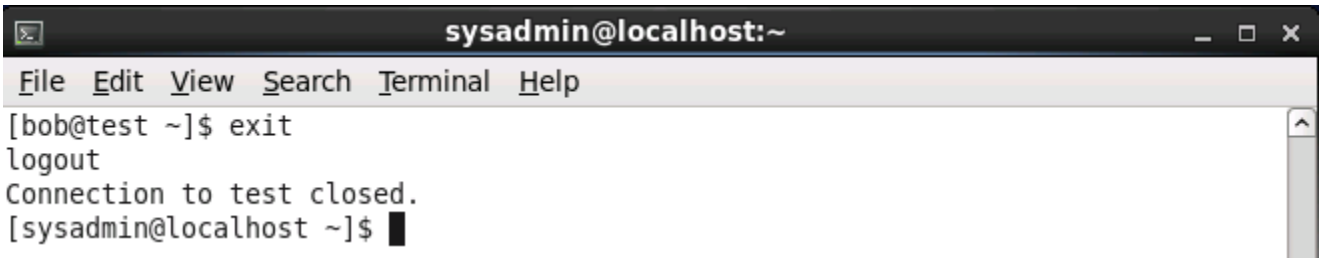
This error could indicate that a rouge host has replaced the correct host. Check with the administrator of the remote system. If the system was recently reinstalled, it would have a new RSA key and that would be causing this error.

In the event that this error message is due to a remote machine reinstall, you can remove the`~/.ssh/known_hosts` file from your local system (or just remove the entry for that one machine) and try to connect again:



12.7.6.2 Returning to the Local Machine

To return back to the local machine, use the `exit` command:



Be careful, if you use the `exit` command too many times, you will close the terminal window that you are working in!