15.4.2 Remove Unneeded Services and Scan Ports

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

Remove Unneeded Services and Scanning Ports 0:00-0:37

As a Linux administrator, you need to pay special attention to the security of your system. One of the key things you can do is look for open ports and unneeded services. To do this, let's first switch to our root user account.

Understand that most Linux distributions include a wide variety of packages by default during the installation process. Most of these packages are necessary. They need to be there in order for the system to run, but there are some that are not necessary. Therefore, one of the first things you should do after you install a new system is to check and see what services are actually enabled and running on your system.

systemctl -a Command 0:37-2:06

To do this we use the 'systemctl' command and we use the '-a' option to list all services. The first part of the output deals with our storage devices. We're not really concerned with those at this point. What we want to look at are all of our service files. What we want to do is scroll through the list of services listed, paying attention to these two columns right here, whether that service is active and whether or not it is currently running.

For example, down here we can see that we have the iscsi service installed. The packages are installed on the system, but it's currently not enabled. It's inactive and it's not running. It's in a dead state.

Any service whose package is installed but is in an inactive state and not running is of less concern. Of more concern are those that are in an active state and are running, such as the CUPS printing service. Right now the CUPS printing service is active and it is running, so you would need to decide, "Do I really need this service running on my system?" If so great--leave it running. If not, though, you may want to consider at a minimum disabling the service, using the systemctl disable command.

For example, I could type, 'systemctl disable cups', or you could actually uninstall the package completely from the system, using the rpm or the YUM command. Basically, if there's a service here that's active and running and you're not familiar with it, go on the web and do a little research. Find out what it does. Do the research necessary to see if that service is really needed on your system.

nmap Utility 2:06-3:34

Another thing you can do is scan for open ports on your system, and one way to do this is to use the nmap utility. nmap is extremely useful. In fact, it can do a lot of other things besides just scan for open ports. For example, it can capture packets and it can analyze network traffic as well.

An open port on your system is of concern, because it indicates a service running on your system is listening for network requests. Basically, it's a hole into your system from a security standpoint, so it's vitally important to know what services are running on your system that have open network ports. Then you can decide whether or not that port is necessary. Is the service listening on that port necessary for the running of the system, or is it something we can turn off?

Once you find an open port, you need to do some research on the web and find out if it is truly necessary. If not, then go ahead and either shut off the service or uninstall its package completely.

In order to use nmap to run a port scan, we first need to see if it's installed. Let's run 'rpm-qi nmap', and we see that it's not installed. Some distributions will install nmap by default. Others will not. This one didn't, so we'll use the 'yum install' command to install 'nmap'. Yes, we want to download and install the packages. Okay, with nmap installed, I now can use it to run a port scan. I can run a port scan for an open TCP port and I can run a port scan for open UDP ports.

-s Option 3:34-4:58

Let's do a TCP port scan first. To do this I enter 'nmap', and then I specify '-s' followed by a 'T' to indicate we want to perform a TCP port scan. Then I have to specify the IP address of the host that I want to perform the scan on, because nmap can scan both the local system as well as other systems on the network, as we'll see in just a second.

Let's just do a port scan of the local host first, and we see that we have one TCP port currently open--Port 631. This is used by the IPP protocol, which in turn is used by the CUPS printing service on this system, so I would have to decide at this point, "Is this something that I want opened or not?" In this case, probably so, because I'm using it for CUPS printing. I've got a shared printer connected to the system. I want other network hosts to be able to send print jobs to it, so this is okay.

Let's do the same thing, but scan for open UDP ports. I'll change the 'T' to a 'U' to specify that we want to scan for UDP ports this time, and we see that there are a couple of more ports available. We have, for example, a DHCP client running that's using Port 68. We also have the NTP protocol that's using UDP Port 123. Again, I would need to decide, "Do I want these protocols running, yes or no?" In this case, yeah, I need my DHCP client in order to get my IP addressing information when the system boots up and I also need the NTP protocol to synchronize time over the network, so everything is good here.

Port Scan of Remote System Using nmap 4:58-6:09

You can also use nmap to perform a port scan of a remote system as well. In fact, this is a common use for nmap--both by the good guys and the bad guys. You can use nmap from a security administrator's standpoint to check the security of hosts on your system. It can also be used by attackers to check the security of hosts on your system.

In this case, we're going to perform a port scan of a remote host with an IP address of 10.0.0.3. We'll just perform a basic TCP scan. As you can see, there are a lot of services running on that system who have open ports, which again, is a potential security exposure. We see that the SSH daemon is running, so we can establish SSH sessions with that system. We have a web server running. We have an LDAP server running. We have the IPP protocol for CUPS printing running. We have some VNC sessions running.

If I were an attacker, that's what I would zero in on, because I know that I can use the VNC protocol to try to connect to that system and gain access to his desktop. Once again, if I were the administrator of this system, I would look and see whether or not I need all of these ports open on the system. And I might want to go in and close down the VNC service, kind of lock the system up a little bit.

netstat Utility 6:09-6:30

The next utility we want to look at is the netstat utility. netstat is another very useful tool. It can be used to list open network connections, display your routing table, and display information about the network interface in your system. The syntax for using this command is to enter 'netstat', followed by the option that you want to use to view specific information.

-a Option 6:30-7:03

For example, if I wanted to view a list of all listening and non-listening sockets on my Linux system, I would use the '-a' option. Remember, a socket represents one end of a network communication session, and it can be in several different states. The socket could be listening, in which case no active connection exists, but it's waiting for one to occur; or it could be active, where a connection does exist, and data is being transferred back and forth to the other end of the communication session to a socket on a different system.

This particular system actually doesn't have much going on.

SSH 7:03-9:08

I want to SSH to a different system. We're going to go look at this one over here, 10.0.0.3, because as you can see, it does have a lot more going on. It's a server system. To do this, I'll enter 'ssh 10.0.0.3 -l', specify the user account I want to connect as. I'm going to do a very bad thing and connect over the SSH connection as the root user on that other system. We'll accept security key, and I have to enter the password for my root user on that other system.

You can see from the prompt down here that I'm currently logged in to that remote system, and I'm executing commands over there instead of on this system over here. Let's run 'netstat -a', and I'm going to pipe (|) the output to 'more', because it will be quite long. Here you can see a list of all the sockets that have been established on this system and what state they're in. You can see that are some are in a listening state and some are in an established active state. For example, you can see that the LDAP server running on this system has established a socket, but currently nothing is connected to it on the other end, so it's just idle. It's in a listening state.

I'm going to actually quit out of here and I'm going to run the 'netstat' command again, '-a', and I'm going to look specifically for the SSH socket. The reason for that is because we're using that socket right now, because we've established an SSH connection with this remote system. Here you can see that session. Here's the SSH protocol and here's the IP address, the foreign address, that's connected to it, and that's the address of this system over here--10.0.0.136--and it's in an established state. In other words, a communication session has been set up between the socket on my system here and the socket on this system over here.

Once again, you would want to review the information displayed by the netstat -a command and see if there are any sockets that have been established that really shouldn't be. Again, if you don't know what a socket does, take a look at the information in the output of the netstat -a command and do your research on the web to find out if that's the necessary service or not. Let's go ahead and get out of our SSH session. We'll 'exit'. We're back now at the console of our local system.

-I Option 9:08-9:23

Another thing you can use the netstat command to do is view statistics for your network interface. We run 'netstat -I' to do this, and when we do, we receive statistics for received packets, and we also see statistics for transmitted packets over here.

-s Option 9:23-9:47

You can also use the 'netstat' command to view summary information for each of the protocols on the system. We enter 'netstat -s', and again I'm going to pipe (|) the output to more. Here at the top of the output you can see summary information for each protocol configured on the system. We have IP, ICMP, TCP. Go down. We'll see UDP and so on. Go ahead and 'exit' out.

-r Option 9:47-10:10

The last thing I want to show you is how to view the routing table using the 'netstat' command. To do this we use the '-r' option. And when we do, we see all of the different routes that have been configured on the system. For example, we have the default gateway route. If an outgoing packet from this system is being addressed to a system that does not reside on the current network, then by default it's going to be sent to the router with this IP address: 10.0.0.254.

Summary 10:11-10:22

That's it for this demonstration. In this demo we talked about how to increase the security of your system by checking for open ports and unneeded services on your system. We did this using the systemctl command, the nmap command, and the netstat command.