15.5.2 Configure xinetd and TCP Wrappers

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

Configure xinetd and TCP Wrappers 0:00-0:14

In this demonstration, we're going to discuss how to configure the xinetd super daemon, and also configure the services that are managed by xinetd, using TCP Wrappers.

---

xinetd Super Daemon 0:15-3:25

As you know, the xinetd daemon is a super daemon. It starts other services. The xinetd daemon uses the files in the '/etc/xinetd.d' directory. We do an 'ls' command here. We see that there is one file for each service managed by xinetd.

It's important to note that these files here in this directory do not configure the service itself, they just tell xinetd how to start each service. For example, let's take a look at the time file.

We'll do a 'cat' command, 'time'. These parameters that you see within this file tell xinetd how to start the time service. For example, it specifies what protocol to use. In this case, it's going to use TCP.

I should point out that there are two different versions of the time file. There is the standard time file, which uses TCP. But there's a second version up here that you could use that uses UDP, instead of TCP.

Notice that in the TCP version of the file, the wait parameter is set to no.

If we were to use the UDP version of the file, the wait parameter would be set to yes. In addition, it tells us what user account to run the service as. Notice here though, by default, it's going to run as the root user. That's a very bad security practice. If I were to deploy this service in a production environment, first I would go and create a dedicated system user account for the time service, and then I would edit this file and run this service as the time service account, not as the root account. For what we're doing today, this will work just fine.

Also notice the disable parameter right here, and that it is set to yes. When disable is set to yes, it means that xinetd is not allowed to start this service. If we want xinetd to be able to start this service, we need to actually change this and set it to a value of no.

Let's go ahead and do that right now. 'vi time', press the Insert key, Escape, and we will save our changes. We've configured the time file such that xinetd is allowed to start it, but xinetd actually doesn't know that yet.

In order to implement the change that we just made in this file, we actually have to restart the xinetd daemon.

Let's do 'systemctl restart xinetd'. Let's take a look at the status of xinetd now. It looks like everything worked great. The daemon is active and running, and there are no error messages shown in the output here.

At this point, if we wanted to really allow network hosts to connect to our system, and have xinetd start the time service and have the remote host access that time service, we would actually have to go into our firewall configuration as well and open up whatever ports are used by the time service.

TCP Wrappers 3:26-7:28

At this point, we need to shift gears a little bit and talk about how to secure these services that are managed by xinetd. Basically, we want to be able to say who is allowed to access these services and who is not allowed to access these services.

This is done using TCP Wrappers. With TCP Wrappers, we can create configuration files that specify which users are allowed to access these services and which aren't. In order to do this, we first have to have the tcpd package installed on the system.

Let's take a look and see if it is. 'rpm -qi tcpd', and we see that it is not installed. We're going to go ahead and install it.

I am working on a SUSE Linux system and not a Fedora system, so I can't use the YUM command. YUM is not implemented on SUSE Linux. Instead, we use a different command that basically functions in exactly the same way called zypper.

We do 'zypper install tcpd'. zypper will go out to my installation repositories on the internet, resolve any dependencies, pull down the necessary packages, and install them on my system for me.

'Yes' to go ahead and install. Okay, so at this point tcpd is installed on my system. We can verify that by running the 'rpm' command again, and we see that it is in fact installed now.

Because we have enabled the time service, let's go ahead and secure it as well.

Before we do that though, we need to know two critical things. The first thing we need to know is the path of the tcpd executable that we just installed. To do that, we can run the 'which' command 'tcpd', and it tells us that the TCP daemon's executable is in /user/sbin.

The second thing we have to know is the path to the time executable that the xinetd daemon's actually going to run. Let's run the same command again 'which', and this time look for 'time'. You can see that it's located in /user/bin.

With this information in hand, let's go back and edit our time service file again and set it up to use TCP Wrappers. Press the Insert key. Go down here and add a new line.

What we want to do is reconfigure xinetd such that if a request for the time service is received by the system, we want xinetd to start TCP Wrappers now instead of the time service. We want it to run the tcpd daemon.

To do this, we add a new line to the file named 'servers', put an '=', and instead of running the time daemon, we're going to instead run '/usr/sbin/tcpd'. Recall we used the which command to find out where the TCP daemon resided. This is the path that we came up with.

If a request for the time daemon is received, xinetd instead of actually starting time, is instead going to start TCP Wrappers. Once that happens, we have to tell TCP Wrappers where the actual time daemon exists.

We do that using the 'server_args' parameter. We have to set that equal to the path to the time daemon itself '/usr/bin/time'. When a request for the time service is received, the TCP daemon will be run by xinetd, and then tcpd will in turn load the time service.

Go ahead and save our changes and 'exit' the file. We do need to restart the xinetd daemon at this point to make the changes we just implemented take effect.

Access Restriction with Time Daemon 7:29-10:58

Now that TCP Wrappers have been implemented, we need to configure the access controls the TCP Wrappers will use in order to restrict access to the time daemon. This is done using two files in the /etc directory.

Let's use the 'ls' command to view the host files in the /etc directory. The two that we're going to be concerned with here are /etc/host.allow and /etc/host.deny.

The host.allow file specifies which network hosts are allowed and the host.deny file specifies which hosts are not allowed. It's important to note here that this file host.allow is actually read first, and any rules it contains are applied before host.deny.

Here's the key thing. Within each of these files, if tcpd finds a matching rule to the current request, then the search is stopped and all the remaining rules are ignored. Because host.allow is read first, if we have an allow rule in here for a particular host, and the current request matches that rule, then that rule will be applied and it will not consult host.deny.

That means we could have an allow rule for a particular host in here and also a parallel deny rule for that very same host in this file, but because this file is processed first, the allow rule will be applied and the deny rule will not be applied. Just remember the processing order of these two files. That's very important.

Let's suppose we want to restrict access to the time service on this system to a network host that has an IP address of 10.0.0.161. To do this, we would enter 'vi /etc/host.allow', and press the Insert key, and we would scroll clear down to the bottom.

The syntax for this file is to simply specify the service name, followed by the allowed IP address. In this case, the name of the service is 'time', enter ':', and then I enter the allowed IP address. In this case it'll be '10.0.0.161'. If a host named 10.0.0.161 tries to connect to this host, it will be allowed.

Notice in these examples up here that I could use a different value. I could specify ALL if I wanted to. If I entered ALL, that would allow all network hosts to be able to use the time daemon.

I could also specify a subnet if I wanted to. I could enter in a network address and a subnet mask to allow hosts just on a particular subnet to access the daemon. In this case, we're just going to allow one single host, and we'll specify 'ALLOW'.

Okay. We'll 'exit' the file, save our changes. If another host tries to connect to the time service, it's going to go through host.allow and it won't find a match, so it's then going to process host.deny. I actually need to edit that file as well, and we need to specify that everybody else is denied. We enter 'time: ALL : DENY'.

We've now configured TCP Wrappers such that only one host will be allowed to start the time service on this system.

Summary 10:59-11:08

That's it for this demonstration. In this demo we talked about how to configure the xinetd super daemon, and we also implemented security for all those services managed by the xinetd super daemon, using TCP Wrappers.
