15.5.3 The xinetd Daemon and TCP Wrapper Facts

The Extended Internet Service Daemon (xinetd) is a super-server daemon that runs on many Linux distributions. It manages many network services on a Linux system.

This lesson covers the following topics:

- Key xinetd facts
- Configuration files
- Steps to use TCP wrappers to restrict daemon access

Key Facts

Be aware that the xinetd daemon:

- Starts and stops network daemons as necessary to provide port security and conserve resources.
- Receives requests for network services from client computers.
- Acts as an intermediary between the user requesting a network service and the actual daemon that provides the service.
- Can be configured to grant and deny access to specific services based on the IP address of the computer making the request. This is done using a separate package called TCP wrappers.
- Increases server latency and might not be optimal for servers with very high request volumes.
- Must be restarted after configuration changes.

The inetd daemon is a super-daemon (similar to xinetd) that was used on older Linux distributions. Like xinetd, the inetd daemon acts as a mediator for connection requests to network services running on the Linux host. It accepts connection requests from client systems, starts the requested service, and the forwards the requests from clients to the newly started daemon. When the transaction is complete and the connection from the client is terminated, the daemon is stopped on the Linux host.

Unlike the xinetd daemon, all of the services managed by inetd are configured using a single configuration file (/etc/inetd.conf). Each line in this file configures a single service to be managed by inetd using the following syntax:
service_name socket_type protocol flags user executable arguments

Configuration Files

Use the following files to configure the xinetd super daemon:

File	Description
/etc/xinetd.conf	The **/etc/xinetd.conf** file configures the xinetd daemon. The default configuration for this file rarely needs adjustment; however, be aware of the following parameters: - **instances** sets the maximum number of concurrent requests xinetd can support. - **log_type** configures the location where xinetd writes logs to. The default is the **/var/log/xinetd.log** file. - **log_on_success** determines whether successful connections are logged. - **log_on_failure** determines whether failed or disallowed connections are logged. - **cps** limits the number of connections per second. - **includedir /etc/xinetd.d** tells the xinetd daemon to use the service-specific configuration files in the **/etc/xinetd.d** directory.
/etc/xinetd.d	The **/etc/xinetd.d** directory contains a file for each network daemon managed by xinetd. The configuration file determines how xinetd will enable the network daemon. Parameters include: - **disable** enables and disables the daemon. - **service** names the daemon. The name often comes from the **/etc/services** file. - **socket_type** determines whether the socket type is a stream. - **wait** specifies whether the daemon is single-threaded or multi-threaded. A Yes specifies single-threaded. - **user** determines the user under which the daemon runs. - **server** lists the path to the daemon's executable. - **log_on_failure** defines logging specifications for failed logins. Each enabled daemon requires an exception in the host-based firewall to open the port for that daemon.

Steps to Use TCP Wrappers to Restrict Daemon Access

TCP wrappers (tcpd) use the IP addresses of incoming network packets to allow or deny access to computers or daemons. Xinetd can use TCP wrappers to restrict access to enabled daemons. To use TCP wrappers with xinetd, consider the following steps:

Step	Description	Examples
Install or verify that the TCP wrappers package is installed.	Ensure that the TCP wrappers package (**tcpd**) is installed with the **rpm -q** or **dpkg -d** commands. If not, use **dnf**, **zypper**, or **apt-get** to install it.	**rpm -q tcpd** uses the **rpm** utility to determine whether tcpd is installed. **dpkg -s tcpd** performs the identical function on Debian distributions.
Edit the daemon files in **/etc/xinetd.d**.	Edit and save the **/etc/xinetd.d** daemon file(s) for the daemon(s) as follows: ▪ Comment out the existing **server=** line with the number symbol (#). ▪ Add the line **server = /usr/sbin/tcpd** to send requests through tcpd so it can grant or deny access. ▪ Add the line **server_args = *path_to_executable*** to provide the tcpd daemon with the path to the executable file of the service. ▪ Set the disable line to **no**.	**# server = /usr/bin/rsync** tells the computer to treat this line as a comment and ignore it. **server = /usr/sbin/tcpd** replaces the direct path to the executable with the path to tcpd so the request can be filtered. **server_args = /usr/bin/rsync** specifies the executable to be started if access is granted. **disable = no** enables the service through xinetd.d.
Restart xinetd.	Restart the xinetd to enforce the changes made to the **/etc/xinetd.d** daemon file(s).	**service inetd restart** restarts the daemon on computers that use the path specified. Some distributions place the daemon in another location.
Modify **tcpd** control files.	Modify the following **tcpd** control files to determine which computers can access the services: ▪ **/etc/hosts.deny** denies services to the specified host(s) or subnets. ▪ **/etc/hosts.allow** permits services to the specified host(s) or subnets. Be aware of the following details: ▪ The **/etc/hosts.allow** file is read first and applied before **/etc/hosts.deny**. ▪ In each of these files, the search is stopped and all remaining rules are ignored if **tcpd** finds a matching rule. Both files have the following syntax: ▪ Use **service: ipaddresses** to specify the host(s). ▪ Use ***service*: *subnet*** to specify a subnet.	**ALL: 192.168.0.0/255.255.255.0** specifies all computers on the 192.168.*.* network. **ALL** specifies all services. The subnet mask follows the network. **ftp: 192.168.10.10** specifies FTP access for only the computer with the IP address of 192.168.10.10 **sshd: 192.168.** specifies sshd access for all computers on the 192.168.*.* network. **sshd: ALL** specifies sshd access for all computers. **sshd: ALL EXCEPT fs1** specifies sshd access for all computers but fs1.
Confirm TCP wrapper configuration.	Use **tcpdchk** to test and display any potential or real problems with the TCP wrapper configuration. **tcpdchk** compares the **/etc/hosts.deny** and **/etc/hosts.allow** files against the configuration files.	