16.4. xinetd Configuration Files

The configuration files for xinetd are as follows:

- /etc/xinetd.conf — The global xinetd configuration file.
- /etc/xinetd.d/ directory — The directory containing all service-specific files.

16.4.1. The /etc/xinetd.conf File

The /etc/xinetd.conf file contains general configuration settings which effect every service under xinetd's control. It is read once when the xinetd service is started, so for configuration changes to take effect, the administrator must restart the xinetd service. Below is a sample /etc/xinetd.conf file:

```
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}
includedir /etc/xinetd.d
```

These lines control the following aspects of xinetd:

- instances — Sets the maximum number of requests xinetd can handle at once.
- log_type — Configures xinetd to use the authpriv log facility, which writes log entries to the /var/log/secure file. Adding a directive such as FILE /var/log/xinetdlog would create a custom log file called xinetdlog in the /var/log/ directory.
- log_on_success — Configures xinetd to log if the connection is successful. By default, the remote host's IP address and the process ID of server processing the request are recorded.

- `log_on_failure` — Configures `xinetd` to log if there is a connection failure or if the connection is not allowed.
- `cps` — Configures `xinetd` to allow no more than 25 connections per second to any given service. If this limit is reached, the service is retired for 30 seconds.
- `includedir` `/etc/xinetd.d/` — Includes options declared in the service-specific configuration files located in the `/etc/xinetd.d/` directory. Refer to Section 16.4.2 *The `/etc/xinetd.d/` Directory* for more information about this directory.

> **Note**
>
> Often, both the `log_on_success` and `log_on_failure` settings in `/etc/xinetd.conf` are further modified in the service-specific log files. For this reason, more information may appear in a given service's log than the `/etc/xinetd.conf` file may indicate. Refer to Section 16.4.3.1 *Logging Options* for more about logging options.

16.4.2. The `/etc/xinetd.d/` Directory

The files in the `/etc/xinetd.d/` directory contains the configuration files for each service managed by `xinetd` and the names of the files correlate to the service. As with `xinetd.conf`, this file is read only when the `xinetd` service is started. For any changes to take effect, the administrator must restart the `xinetd` service.

The format of files in the `/etc/xinetd.d/` directory use the same conventions as `/etc/xinetd.conf`. The primary reason the configuration for each service is stored in a separate file is to make customization easier and less likely to effect other services.

To gain an understanding of how these files are structured, consider the `/etc/xinetd.d/telnet` file:

```
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = yes
}
```

These lines control various aspects of the `telnet` service:

- `service` — Defines the service name, usually one listed in the `/etc/services` file.
- `flags` — Sets any of a number of attributes for the connection. `REUSE` instructs `xinetd` to reuse the socket for a Telnet connection.
- `socket_type` — Sets the network socket type to `stream`.
- `wait` — Defines whether the service is single-threaded (`yes`) or multi-threaded (`no`).
- `user` — Defines what user ID the process process will run under.
- `server` — Defines the binary executable to be launched.
- `log_on_failure` — Defines logging parameters for `log_on_failure` in addition to those already defined in `xinetd.conf`.
- `disable` — Defines whether or not the service is active.

16.4.3. Altering `xinetd` Configuration Files

There are a large assortment of directives available for `xinetd` protected services. This section highlights some of the more commonly used options.

16.4.3.1. Logging Options

The following logging options are available for both `/etc/xinetd.conf` and the service-specific configuration files within the `/etc/xinetd.d/` directory.

Below is a list of some of the more commonly used logging options:

- `ATTEMPT` — Logs the fact that a failed attempt was made (`log_on_failure`).
- `DURATION` — Logs the length of time the service is used by a remote system (`log_on_success`).
- `EXIT` — Logs the exit status or termination signal of the service (`log_on_success`).
- `HOST` — Logs the remote host's IP address (`log_on_failure` and `log_on_success`).
- `PID` — Logs the process ID of the server receiving the request (`log_on_success`).

- `USERID` — Logs the remote user using the method defined in RFC 1413 for all multi-threaded stream services (`log_on_failure` and `log_on_success`).

For a complete list of logging options, consult the `xinetd.conf` man page.

16.4.3.2. Access Control Options

Users of `xinetd` services can choose to use the TCP wrappers hosts access rules, provide access control via the `xinetd` configuration files, or a mixture of both. Information concerning the use of TCP wrappers hosts access control files can be found in Section 16.2 *TCP Wrappers Configuration Files*.

This section discusses using `xinetd` to control access to services.

Note

Unlike TCP wrappers, changes to access control only take effect if the `xinetd` administrator restarts the `xinetd` service.

Also, unlike TCP wrappers, access control through `xinetd` only affects services contrlled by `xinetd`.

The `xinetd` hosts access control differs from the method used by TCP wrappers. While TCP wrappers places all of the access configuration within two files, `/etc/hosts.allow` and `/etc/hosts.deny`, `xinetd`'s access control is found in each service's configuration file within the `/etc/xinetd.d/` directory.

The following hosts access options are supported by `xinetd`:

- `only_from` — Allows only the specified hosts to use the service.
- `no_access` — Blocks listed hosts from using the service.
- `access_times` — Specifies the time range when a particular service may be used. The time range must be stated in 24-hour format notation, `HH:MM-HH:MM`.

The `only_from` and `no_access` options can use a list of IP addresses or host names, or can specify an entire network. Like TCP wrappers, combining `xinetd` access control with the enhanced logging configuration can increase security by blocking requests from banned hosts while verbosely recording each connection attempt.

For example, the following `/etc/xinetd.d/telnet` file can be used to block Telnet access from a particular network group and restrict the overall time range that even allowed users can log in:

```
service telnet
{
        disable         = no
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        no_access       = 10.0.1.0/24
        log_on_success  += PID HOST EXIT
        access_times    = 09:45-16:15
}
```

In this example, when client system from the 10.0.1.0/24 network, such as 10.0.1.2, tries to access the Telnet service, it receives a message stating the following message:

```
Connection closed by foreign host.
```

In addition, their login attempts are logged in `/var/log/secure` as follows:

```
May 15 17:38:49 boo xinetd[16252]: START: telnet pid=16256 from=10.0.1.2
May 15 17:38:49 boo xinetd[16256]: FAIL: telnet address from=10.0.1.2
May 15 17:38:49 boo xinetd[16252]: EXIT: telnet status=0 pid=16256
```

When using TCP wrappers in conjunction with `xinetd` access controls, it is important to understand the relationship between the two access control mechanisms.

The following is the order of operations followed by `xinetd` when a client requests a connection:

1. The `xinetd` daemon accesses the TCP wrappers hosts access rules through a `libwrap.a` library call. If a deny rule matches the client host, the connection is dropped. If an allow rule matches the client host, the connection is passed on to `xinetd`.

2. The xinetd daemon checks its own access control rules both for the xinetd service and the requested service. If a deny rule matches the client host the connection is dropped. Otherwise, xinetd starts an instance of the requested service and passes control of the connection to it.

Important

Care should be taken when using TCP wrappers access controls in conjunction with xinetd access controls. Misconfiguration can cause undesired effects.

16.4.3.3. Binding and Redirection Options

The service configuration files for xinetd support binding the service to an IP address and redirecting incoming requests for that service to another IP address, hostname, or port.

Binding is controlled with the bind option in the service-specific configuration files and links the service to one IP address on the system. Once configured, the bind option only allows requests for the proper IP address to access the service. This way different services can be bound to different network interfaces based on need.

This is particularly useful for systems with multiple network adapters or with multiple IP addresses configured. On such a system, insecure services, like Telnet, can be configured to listen only on the interface connected to a private network and not to the interface connected with the Internet.

The redirect option accepts an IP address or hostname followed by a port number. It configures the service to redirect any requests for this service to the specified host and port number. This feature can be used to point to another port number on the same system, redirect the request to different IP address on the same machine, shift the request to a totally different system and port number, or any combination of these options. In this way, a user connecting to certain service on a system may be rerouted to another system with no disruption.

The xinetd daemon is able to accomplish this redirection by spawning a process that stays alive for the duration of the connection between the requesting client machine and the host actually providing the service, transferring data between the two systems.

But the advantages of the bind and redirect options are most clearly evident when they are used together. By binding a service to a particular IP address on a system and then redirecting requests for this service to a second machine that only the first machine can see, an internal system can be used to provide services for a totally different network. Alternatively, these options can be used to limit the exposure of a particular service on a multi-homed machine to a known IP address, as well as redirect any requests for that service to another machine specially configured for that purpose.

For example, consider a system that is used as a firewall with this setting for its Telnet service:

```
service telnet
{
        socket_type             = stream
        wait                    = no
        server                  = /usr/sbin/in.telnetd
        log_on_success          += DURATION USERID
        log_on_failure          += USERID
        bind                    = 123.123.123.123
        redirect                = 10.0.1.13 23
}
```

The `bind` and `redirect` options in this file ensures that the Telnet service on the machine is bound to the external IP address (123.123.123.123), the one facing the Internet. In addition, any requests for Telnet service sent to 123.123.123.123 are redirected via a second network adapter to an internal IP address (10.0.1.13) that only the firewall and internal systems can access. The firewall then send the communication between the two systems, and the connecting system thinks it is connected to 123.123.123.123 when it is actually connected to a different machine.

This feature is particularly useful for users with broadband connections and only one fixed IP address. When using Network Address Translation (NAT), the systems behind the gateway machine, which are using internal-only IP addresses, are not available from outside the gateway system. However, when certain services controlled by `xinetd` are configured with the `bind` and `redirect` options, the gateway machine can act as a proxy between outside systems and a particular internal machine configured to provide the service. In addition, the various `xinetd` access control and logging options are also available for additional protection.

16.4.3.4. Resource Management Options

The `xinetd` daemon can add a basic level of protection from a Denial of Service (DoS) attacks. Below is a list of directives which can aid in limiting the effectiveness of such attacks:

- `per_source` — Defines the maximum number of instances for a service per source IP address. It accepts only integers as an argument and can be used in both `xinetd.conf` and in the service-specific configuration files in the `xinetd.d/` directory.
- `cps` — Defines the maximum of connections per second. This directive takes two integer arguments separated by white space. The first is the maximum number of connections allowed to the service per second. The second is the number of seconds `xinetd` must wait

before re-enabling the service. It accepts only integers as an argument and can be used in both `xinetd.conf` and in the service-specific configuration files in the `xinetd.d/` directory.

- `max_load` — Defines the CPU usage threshold for a service. It accepts a floating point number argument.

There more resource management options available for `xinetd`. Refer to the chapter titled ***Server Security*** in the ***Red Hat Enterprise Linux Security Guide*** for more information. Also consult the `xinetd.conf` man page.

Prev | Home | Next
`xinetd` | Up | Additional Resources