4.3.5 Boot Target Facts

The systemd daemon uses boot targets to set the system state. A boot target represents one of several different modes that the Linux system can be configured to run in.

Earlier versions of Linux that were based on the init daemon used runlevels instead of boot targets to set the system state.

This lesson covers the following topics:

- Boot target files
- Equivalent runlevel names
- systemctl command

Boot Target Files

The default boot target specifies the state that your system will boot into each time it is turned on. In addition, the system state can be dynamically switched between boot targets while the operating system is running. Each boot target is represented by a file (called a target unit) in **/usr/lib/systemd/system**:

Target File	Description
poweroff.target	Halts the system.
rescue.target	Configures the system to run in single-user mode with a text-based user interface. This target sets up a base system and opens a rescue shell for troubleshooting system problems.
multi-user.target	Configures the system to run in multi-user mode with a text-based user interface. This target is commonly used as the default mode for server systems.
graphical.target	Configures the system to run in multi-user mode with a graphical user interface. This target provides all the services of the multi-user target with the addition of a graphical user interface. This target is commonly used as the default mode for desktop systems.
reboot.target	Reboots the system.
emergency.target	Opens a minimal emergency shell for troubleshooting serious system problems.

Equivalent Runlevel Names

Because init-based distributions have been around for decades, systemd provides additional boot targets that use runlevel-like names to help ease the transition from init to systemd:

Target File	Description
runlevel0.target	Equivalent of **poweroff.target**
runlevel1.target	Equivalent of **rescue.target**
runlevel2.target **runlevel3.target** **runlevel4.target**	Equivalent of **multi-user.target**
runlevel5.target	Equivalent of **graphical.target**
runlevel6.target	Equivalent of **reboot.target**

systemctl Command

The **systemctl** command is used to manage boot targets:

Command	Description	Example
systemctl isolate ***boot_target***	Changes the system state to the specified boot target. Changing boot targets with the systemctl command changes only the current system state. If the system is restarted, it will revert back to the default boot target.	Either of the following commands can be used to switch the system to multi-user

		mode with a graphical user interface: ▪ **systemctl isolate runlevel 5.target** ▪ **systemctl isolate graphical.target**
systemctl get-default	Displays the current boot target.	**systemctl get-default**
systemctl set-default ***boot_target***	Sets the default boot target, which is identified by the **/etc/systemd/system/default.target** file. This file is a symbolic link that points to a target file in /usr/lib/systemd/system that should be used by default when the system starts. This command modifies the target file that the default.target symbolic link points to.	Either of the following commands can be used to set the default boot target to graphical mode: ▪ **systemctl set-default graphical.target** ▪ **systemctl set-default runlevel 5.target**

On older Linux distributions, the **init** command was used to manage runlevels and system daemons. The **/etc/inittab** file was used to set the default runlevel.

You can use the **systemd-analyze blame** command to print a list of running units, listed in the order of time to initialize. Consider that initialization time includes the time a unit must wait for another unit to complete. The **systemd-analyze blame** command does not report results for services that start immediately as indicated by type=simple.
