11.1.6 Viewing Log Files

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

View Log Files 0:00-0:06

In this demonstration, we're going to discuss viewing Linux log files.

Linux Log Files 0:07-0:23

Some Linux log files are text files, and can be viewed with any tech utility like cat, less, more, head, tail, and so on. But others are binary files that can be viewedonly with the appropriate utility. We're going to look at both of those in this lesson. I'm going to first switch to my root user account here.

/var/log Directory 0:24-0:40

Almost all of your Linux log files are stored in the same place--that is the /var/log directory. If I do an 'ls' command you can see there are many different log files and subdirectories containing log files within /var/log.

Boot.log File 0:41-2:02

Let's begin by looking at this one right here, it's called boot.log. Understand that during system boot, many useful log messages are generated. They can be very helpful when you're troubleshooting problems.

However, most modern Linux distributions hide these messages with a nice graphical splash screen during system bootup. In the old days you actually could see them being scrolled across the screen, but not anymore.

All is not lost because all of those messages are saved right here in the boot.log file in /var/log. Be aware that on some distributions this file may be named slightly differently. On distributions such as openSUSE, it is boot.msg for boot messages. It's the same file, contains the same information, it just has a slightly different filename.

The boot.log or the boot.message file is a straight text file. You can actually view it with any text manipulation utility you want.

For example, if I wanted to view the end of the boot.log file to see the last few messages that were generated during the system boot, I could enter 'tail boot.log', and I can see the last couple boot events.

If I want to view it all, I enter 'cat boot.log', and all the log messages that were generated as the system booted up are displayed on the screen.

dmesg Command 2:03-3:20

You can view more detailed bootup messages using the 'dmesg' command. The dmesg command is used to view the kernel ring buffer. I'm going to hit Enter here.

You can see that it provides much more detailed boot information than we saw in the boot.log file. As you can see, it's a quite long file. If necessary, you can type it to more so that you can pause the output one page at a time.

One common thing we do with the dmesg command is actually redirect the output to a file. There may be situations where if you've paid for support for your Linux distribution, and you're engaged with a tech support representative trying to solve a problem, they may ask to see the contents of dmesg.

What we do is we type 'dmesg', and then redirect the output to a file. I'm going to put it in my root user's /home directory, and let's call it 'boot.dmeg'.

If I do a 'cat' command of the 'boot.dmeg' file, in my root user's /home directory, we can see that all the contents have been output. I can take that data and send it off to the tech support rep, and they can analyze it to see what's going on with my system.

wtmp File 3:21-3:59

Let's go back and look at the contents of the /var/log directory. Another log file in here that you need to be familiar with is wtmp, or w-temp. wtmp contains a list of all users who have authenticated to this Linux system.

This file is a binary file. It's not a text file, and therefore you cannot use cat, or less, or tail to view it. Instead, you have to use the last command. When you run 'last', the contents of the wtmp file are displayed on the screen. We'll spend more time working with last in a different demonstration.

---

lastlog File 4:00-5:00

There's also a log file in here called lastlog. The lastlog file contains a log of the last time each user authenticated to the system. This is also a binary file, and so you can't view it with cat, less, tail, or head.

You have to use instead the lastlog command. When I run the 'lastlog' command, the contents of the lastlog file are displayed. We'll spend more time with lastlog again in a different demonstration.

I'm going to 'clear' the screen here. Let's do an 'ls' command again.

On older Linux distributions that used init instead of systemd, there was a log file in here that we found incredibly useful called the messages file, and it contained just about all of the log entries for the entire system. It was a text file that could be viewed with cat, less, tail, and so on.

As you can see, there is no file here named messages anymore. It's no longer used on a distribution that uses the systemd daemon.

---

journald 5:01-5:38

Instead of syslogd, systemd distributions use journald for logging. The main log file is located here in the journal directory. The journal is a binary log file. You can't view it with cat, less, tail, head, or any of those other commands that we used to use on older systems to view the messages file that we just talked about.

Instead, you have to use the journalctl command. When I run 'journalctl', the contents of the journal are displayed on the screen. We'll cover journalctl in more depth in a different demonstration.

---

journalctl with the -f Option 5:39-7:05

However, there is one key thing I want to show you here, and that is the fact that the journal can be really useful in situations where you're trying to troubleshoot a specific problem on the system.

To show you how this works, I'm going to open up a new terminal window over here, and I'm going to switch to root. I'm going to go back to my first window. I'm going to 'clear' this so we can see what we're doing.

I'm going to run 'journalctl' with the '-f' option. The -f option will, first of all, display the last few lines--the most recent entries--in the journal on the screen. Then it will continue to monitor the journal for any new entries that are added, and as they are added, they are displayed on the screen.

We can basically see what's going on with the journal. The problem is if I just run the journalctl command straight, it just displays the current snapshot of the journal. It's not interactive, it's not updated. Things could be changing on the system as we're looking at the journal, and we won't see them reflected.

To fix that, we use the -f option to continuously monitor the journal and display new entries as they're added. I'm going to go ahead and press Enter here, and you can see the last few lines of the journal are displayed.

Notice that we didn't get the cursor back. That's because the journalctl command is actively monitoring the journal, waiting for new entries to be added. Let's slide this over just a hair.

---

logger Command 7:06-9:30

I'm going to go over to my other window, and I'm going to generate a log message. I'll do that with the logger command. The logger command is used to manually add entries to the system log, with either syslogd or journald, doesn't matter. It works with both.

It's usually just used for testing purposes to make sure your logging configuration is working properly. For our purposes, we're going to use it to simulate some error event occurring on the system, and we're going to see that error event pop up over here in the journal.

I'm going to enter 'logger ', and then the message that I want to save to the system log in the journal.

I'm going to send a hypothetical "Critical Error Has Occurred" message to the journal. Hit Enter, and notice that the message that I created with logger is immediately displayed over here in the journal. This is a great way to troubleshoot problems on your system, such as a system service that isn't starting properly.

You run journalctl -f in one window, you start the service in the other window, and then you can see the log messages that are generated as that service starts over here. A lot of times you can find really good information for troubleshooting as you do so.

Go ahead and break out of here. 'clear' again, and run an 'ls' command to view the contents of the /var/log directory again. You'll notice that some system services are configured to save their log messages into a separate file outside of the journal.

For example, we have a log file here for the MySQL daemon. We have a log directory here for our web server running on the system. We have a log file here for our system firewall. There's a directory over here that contains log messages for our cups printing daemon, as well. There's also a directory over here for log messages from the samba daemon running on this system.

Usually these files are just text files, and they can be viewed with standard text-viewing utilities. For example, we can run the 'tail' command to view the 'firewalld' log file, to view the last few entries in that log file.

Summary 9:31-9:53

That's it for this demonstration. In this demo we talked about viewing Linux log files. We looked at the boot.log file. We looked at using the dmesg command. We looked at the wtmp file. We looked at the lastlog file. We talked about the messages file that was used on older distributions. We looked at the journal, and then we looked at the log files that are used by various services on the system that are saved in the /var/log directory.