15.2.4 Monitor User Logins

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

Monitor User Logins 0:00-0:24

In this demonstration, we're going to discuss monitoring user logins. One of your best resources in this regard are the log files that your Linux system maintains for you.

Like a CSI detective, you need to practice and develop experience in order to gain an intuitive sense that lets you know when something looks suspicious in one of these files.

---

wtmp Log File 0:23-1:05

With this in mind, let's take a look at the log files that you can analyze in order to identify any suspicious activities. I do need to switch to my root user account first, and we need to switch to my /var/log directory. Do an 'ls' command. There are several different log files in here that can be very useful.

The first one we need to look at is wtmp. This log file contains a list of all the users who have ever authenticated to the system. Be aware that this file is saved in binary format. That means you can't use cat, less, head, or tail. Nor can you use a text editor like vi in order to view it.

---

last Command 1:06-3:59

Instead you have to use the last command. I just simply type 'last' at the shell prompt, and I am going to actually '|' (pipe) it into the 'more' command because this file can get really long. We're going to want to pause the output one page at a time.

When you do, a list of logins is displayed. You'll notice over here that the logins are displayed from most recent to least recent. The last utility displays the user account name, where they logged in to, where they logged in from, when they logged in--and then if they've logged out, when they logged out. Also, how long they were logged in for. If they're still logged in it says, still logged in.

Let's look at this first login here. The rtracy user, that's me, logged in to :0; that's the graphical desktop right here. Here's when I logged in to the graphical desktop, and I haven't logged out. I'm still logged in. Notice over here that it says I logged in from :0, which basically means I logged in from the local system, at the console of this system.

Notice there's another rtracy session currently still logged in. Instead of logging in to :0, it shows me I'm pts/0. That's this window right here. From within my graphical environment I opened up a terminal session, and I'm logged in as rtracy. You can see that I still have that session open right here.

There was a second window that I opened right here--a second terminal window that I opened from the local system. It was pts/1, but I actually closed that window, effectively logging me out. We can see my logout time right here, and how long I had that window open.

Notice also here that the ksanders user is currently logged in to this system on pts/1. Notice here that she logged in from a different system somewhere else on the network instead of the local system.

When you review this file, you need to look for anything that just appears unusual. For example, a login that occurred at three in the morning. You know that nobody's working at three in the morning, at least not usually, and therefore that might be a little suspicious.

Is it a guarantee that that's an attack? May not be. Could just be somebody who's late on their deadline and is working late at night. Could it also be somebody who's trying to log in when nobody's looking? Definitely. You'd want to check that out.

I'm going to get out of last here. Let's 'clear' the screen and perform the 'ls' command again. There's another file in /var/log called lastlog right here. This file is similar to wtmp, but it's a little bit shorter, because this file contains just a list of all the users in the system, and when they last logged in. Hence the filename of lastlog. Just like with wtmp, you can't view the lastlog file with cat, less, head, or tail, or with a text editor.

---

lastlog Command 4:00-4:39

Instead, you have to use the lastlog utility. You type 'lastlog' at the shell prompt, and a list of all the users on the system is displayed and when they last logged in. For example, the root user is listed here, where they logged in to. They logged in to a terminal window and this is when

they logged in.

You'll notice all the service accounts in your password file are also listed. And because they're system accounts, you can't use them for login. Hence, they list as having never logged in.

Let's go ahead and 'clear' the screen. In addition to your log files, there are a variety of command line tools that you can use to see who is currently using the system.

---

who Command 4:40-5:00

The first one is the 'who' command; who lets you see who is currently logged in to the system. In this example, you can see that the rtracy user is currently logged in, both to the graphical desktop and also to a terminal session. The ksanders user is also logged in to a terminal session from a remote host: opensuse.nebo-tech.com.

---

finger Command 5:01-5:31

In addition to who, you can also use the finger utility to see who's currently logged in to the system. The finger is not installed, so I'm going to have to install it real quick. With some distributions, finger is automatically installed, with others it is not.

Let's 'clear' the screen and run 'finger' again. Here you can see the username. You can see the full name of the user, where they logged in to, the login time, and where they logged in from.

---

fuser Command 5:32-7:08

The last command that we're going to look at in this demonstration is the fuser command. Sometimes it's just called fuser. fuser is useful in situations where you need to find out who is either running a command on your system, or who has a file open on your system.

Again, this is a great way to monitor user logins. To look for things that are out of whack; that someone is doing something that is suspicious. For example, I'm going to open up a new terminal window over here.

As my rtracy user I'm going to run 'top'. At this point, as root, I can find out who currently is running top--who has it open. To do this, I use the 'fuser' command, and I use the '-u' option to identify which user owns the process that we're going to look at.

The name of the executable while it's running the process in question is '/user/bin/top'. Press Enter. It first identifies the process ID of the process for this executable, 6664, and it tells us the name of the user who currently owns that process: the rtracy user.

This little e right here is important. It indicates what the state of that file is. In this case, e indicates that it's in the executable that's being run.

If, on the other hand, we had used fuser and we had specified a word processing file over here instead of an executable, then it would tell us whether or not the user has the file open, with an f. An f right here would indicate that the file is open and being written to.

Summary 7:09-7:21

That's it for this demonstration. In this demo we talked about monitoring user logins. We first looked at the last command. We then looked at the lastlog command. Then we looked at the who command. We looked at the finger command, and then we ended this demonstration by looking at the fuser command.
