15.2.6 Auditing Files

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

Audit Files 0:00-0:59

In this demonstration, we're going to discuss auditing files. Specifically, we're going to look for files within our Linux file system here that are executable files. In other words, that are commands that have the SUID permission assigned and are owned by root.

To do this, I first need to switch to my root user account. These types of files represent a problem from a security standpoint.

If you think about it, when that file is run, because it has the SUID permission set, the process created when the command is run is granted access to the file system based on the user who owns that executable file--not the user who actually ran the command with the shell prompt.

Which basically means that process runs as root. There are some processes that have to be configured that way, but doing so haphazardly on processes that don't need it creates a security risk. We need to verify that those files that have the SUID permission set really do need it.

SUID Permission 1:00-2:11

To find files such as this, we use the 'find' command. We first need to specify where we want to start searching from. We're going to begin at the /root directory and work our way clear down through the entire file system through the directory tree.

Then we need to specify what we're looking for. We're just looking for files, so we enter '-type f'.

Then we need to specify the '-perm' option. When we use the -perm option with the find command, it allows us to look for files that have a specific set of permissions.

We can basically specify what file mode we want to match on. In this case, we're looking for files whose owner has the SUID permission to sign. '-u' stands for the owner. '+s' means we're looking for the SUID permission assigned to the file owner. Then we enter '-ls' to list any matches that find locates on the screen.

It takes just a minute to run because it has to search through the entire file system. Notice that in the output of the command that we did find quite a few files that have the SUID bit set and that are owned by the root user, which you can see right here.

Executable and Writable 2:12-3:46

At this point, we would want to review the list of files over here and make sure they actually need to be configured this way. In addition, we also should look for files that are writable and executable by others. That's kind of an odd combination.

It's not one that you would commonly see. In fact, it could potentially create a security risk, because you're basically allowing anybody on the system to edit the file and also to execute that file.

To find these types of files, we enter 'find'. We'll again start at the root of the file system, at the /flash directory. We're looking for just files, so we use 'type -f', and then when we specify the mode of the file that we want to match on.

In this case, we want to look for all files to whom others have been granted the write permission and also others have been granted the execute permission. We use '-ls' to print the results on the screen. If all goes well, we should actually find no matching files, and that is the case. We didn't come up with any matches.

Because these types of files represent a fairly significant security risk, you really ought to create a cron script that runs both of these commands that we just looked at on a regular basis. Every week or every month or so, and output that information to a log file so you can see if somebody has created a security breech by giving executable and writable permissions to a particular file to others, or that have the SUID permission enabled for files that are owned by root.

Summary 3:47-3:55

That's it for this demonstration. In this demo we talked about auditing files. We first looked for files that are owned by root that have the SUID permission set. Then we looked for files that are executable and writable by others.
