15.8.1 Security-Enhanced Linux

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

Security-Enhanced Linux 0:00-0:37

SELinux, or Security-Enhanced Linux, is an implementation of a mandatory access control mechanism in the Linux kernel. Put simply, it is a security enhancement that allows users and administrators greater control over access.

Using SELinux, you can restrict access to variables, such as which users and applications can access which resources. SELinux is becoming a standard feature on many distributions, including Red Hat, CentOS, and Fedora.

In this lesson, we're going to cover the basic concepts of SELinux; how to manage the program; and the modes, tools, and commands you can use with SElinux.

SELinux Operations 0:38-1:57

Keep in mind that when you use standard Linux access controls, such as granting permission using read, write, and execute (-rwxrw-r--), users and applications can both modify the controls.

On the other hand, with SELinux, access controls are determined by an SELinux policy, which can only be changed by an administrator. This prevents a careless user or misbehaving applications from making permission changes.

In addition, SELinux also adds finer granularity to access controls because SELinux access decisions are based on all available information, such as an SELinux user, role, type, and, optionally, a level.

For example, instead of only being able to specify who can read, write, or execute a file, SELinux lets you specify who can unlink, append only, move a file, and so forth. SELinux also allows you to specify access to many resources other than files, such as network resources and interprocess communication (IPC).

However, with all of these benefits, it's important to understand that SELinux is not antivirus software; it does not replace passwords, firewalls, or other security systems, nor is it an all-in-one security solution.

SELinux is designed to enhance existing security solutions, not replace them.

SELinux is installed and enabled on many Linux distributions, such as Red Hat, CentOS, and Fedora.

SELinux Policies 1:58-2:45

When SELinux is enabled, policies are used to determine how and what items are protected. Policies are a set of rules that guide the SELinux security engine. For example, the SELinux policy describes the access permissions for all users, programs, processes, files, and devices they act upon.

SELinux uses one of two different policies. The first is the Targeted policy. This is the default policy. It applies access controls to certain processes. These targeted processes run in a confined domain, and processes that are not targeted run in an unconfined domain.

The second policy is Multi-Level Security, or MLS. It's not installed by default, but can be installed if needed. MLS prevents unauthorized users from accessing protected data and applications.

SELinux States and Modes 2:46-3:29

SELinux has two modes, Enforcing and Permissive. In enforcing mode, SELinux denies access based on SELinux policy rules. In other words, if something happens on the system that is against the defined policy, the action is both blocked logged. In permissive mode, the SELinux policy is not enforced. This means that SELinux does not block or deny anything from happening; however, it does log anything it would have blocked in Enforcing mode.

Permissive mode is a great tool for when you want to test a Linux system that has never used SELinux and you want to get an idea of any problems you may have or when you need to troubleshoot an issue. The nice thing about these modes is that you can switch between the two as needed without requiring a system reboot.

SELinux Status 3:30-4:05

Before we get into how to manage the different SELinux modes, let's discuss how to determine whether SELinux is installed and what mode SELinux is currently using.

The easiest way to do this is to use the sestatus command. When run from a terminal, as can be seen here, sestatus provides valuable information regarding its status (enabled or disabled), the policy being used (targeted or MLS), and its current mode (enforcing or permissive).

If you only need to know which mode is being used, you can also use the getenforce command. This command will return one of three modes: Enforcing, Permissive, or Disabled.

Changing Modes 4:06-5:07

In most cases, you'll run SELinux in Enforcing mode. But if you want to see how SELinux will function on a new system or you're having issues and need to troubleshoot SELinux, you may want to switch to Permissive mode.

The setenforce command is the quickest way to temporarily switch modes. This command can only run using root permissions and supports only two options, 0 and 1.

For example, if you enter ‘setenforce 0', SELinux will run in Permissive mode. If you run ‘setenforce 1', SELinux will run in Enforced mode.

To change a mode permanently, you'll need to edit the /etc/selinux/config file and change the SELINUX line to the desired mode.

If, for some reason, you feel you need to turn SELinux off, you can set the SELINUX line to Disabled. Just remember that when SELinux is disabled, it's completely turned off and does not log anything.

Note that if you're disabling or enabling SELinux, you must edit the config file and then reboot your system before your changes will take effect.

SELinux Boolean 5:08-6:01

SELinux also gives you the ability to customize a policy by enabling or disabling a set of policy Booleans.

Booleans allow you to change part of the SELinux policy at run time. This lets you change portions of the policy without reloading or recompiling the SELinux policy. And you don't need very much experience with policies to handle booleans.

For example, as you work with SELinux, you may notice that some activities are denied even though there is a good reason to allow them. When this reason depends on certain factors or choices, SELinux policy writers are encouraged to make the policy optional so that it's not enabled by default.

To enforce this type of policy, you will need to use the boolean commands.

To view a list of booleans, you can use 'getsebool -a'.

This command lists the available Booleans and shows the booleans' current state.

Booleans can be one of three states: on, off, or pending.

Changing Booleans 6:02-6:44

You can change the current state of an SELinux boolean using the setsebool command. For example, if you had a boolean named zoneminder_run_sudo, you could change its state from off to on by entering ‘setsebool zoneminder_run_sudo on'.

Likewise, to turn a boolean off, you would use the off switch. You can replace on with 1 or true, and you can replace off with 0 or false. When you use this command as shown, the changed value will take effect immediately. But the next time the computer is rebooted, the

default value will be used.

Therefore, if you want to make the change to the boolean permanent, you need to add the -P switch, as shown here.

SELinux Security Context 6:45-7:23

Before we end, we need to cover SELinux security contexts and how they're managed. A security context, sometimes referred to as a security label, is the method SELinux uses to classify resources, such as processes and files, on a SELinux-enabled system. This context allows SELinux to enforce rules for how and by whom a given resource should be accessed.

A security context is typically shown as a string of three or four words. Each word specifies a different component of the security context, namely the user, role, type, and level of that file or process, and each word is separated by a colon.

---

ls -Z 7:24-7:51

To view the SELinux context for a file, you would run 'ls -Z' and then the name of the file, such as 'ls -Z /user/bin/pkexec'.

With this particular example, you can see that, in addition to the normal read-write-execute permissions, the security context system_u:object_r:bin_t:s0 is shown.

---

ps -Z 7:52-8:15

In addition, if you want to view the security context for all of the active processes, you can use ps -eZ.

Since this will probably produce a lot of information, you can limit the results to only what you're looking for by including the grep command.

For example, if you only want to view the active processes for ibus, you can enter ‘ps -eZ | grep ibus'.

Changing Security Context - chcon 8:16-10:19

From time to time, you may find that you need to change the security context of an object. You can use the ‘chcon' command to do this. chcon stands for Change Context.

There are many ways to use this command. For example, you can change the full security context, meaning every aspect of the security context.

You can change the context using another file as a reference, or you can change just a specific portion of the context, such as the type. And if you need to make a lot of changes, you could even change the security context recursively. This is only a small list, so refer to you man pages for details about all of the available options.

Let's look at a few examples of how you can use this command.

First let's see how you can change the full SELinux context.

If I had a file named file1 in the Sales directory, I would first determine its current security context using the ls -Z command.'ls -Z file1' shows me the current security context for file1.

In the output, I notice that the type should be user_home instead of admin_home.

To change this, I can use the chcon command to specify the full context by typing ‘chcon unconfied_u:object_r:user_home_t:s0 file1'.

And now, if re-run 'ls -Z file1', you see that the context has changed.

This works fine. But since the only thing I needed to change was the type, I could have just changed an individual component using the -t switch.

In a similar fashion, you can use -u to change the user field and -r to change the role field.

You can use the -R switch to make the same change to all the files in a single directory.

For example, if I have a directory named Sales, and it contains several files that all need the type component changed, I'd type ‘chcon -R -t user_home_t Sales'. "user_home_t" is the new type component.

Restoring Security Context 10:20-10:50

The last command I want to show you is restorecon. Restorecon is most commonly used to restore SELinux security contexts back to their default values.

As with the chcon command, you can do this one file at a time, or for all the files in a directory.

For example, if I only wanted to restore the security context for one file, I would run ‘restorecon file1'.

However, if I wanted to restore the security context for all the files in a directory, I'd run ‘restorecon -R' for recursive and then the folder name.

Summary 10:51-11:04

That's it for this lesson.

In this lesson, we covered the basic concepts of SELinux. Then we talked about the different modes in which SELinux functions.

We talked about profiles and explained their role in implementing SELinux.

And finally, we showed you several commands used to manage SELinux.
