

Content

Objectives

1  
1.1  
1.2  
2  
2.1  
2.2  
3  
4  
5  
**6**  
7  
8  
9  
10  
11  
11.1  
12  
13  
14  
14.1  
14.2  
15  
16  
17  
18  
19  
20  
21  
22  
22.1

## Administrative Access



## Administrative Access

There are many Linux commands which deal with sensitive information like passwords, system hardware, or otherwise operate under other exceptional circumstances. Preventing regular users from executing these commands helps to protect the system. Logging in as the root user provides administrative access, allowing for the execution of some of the privileged commands.

### The `su` Command

```
su OPTIONS USERNAME
```

The `su` command allows you to temporarily act as a different user. It does this by creating a new shell. The shell is simply a text input console that lets you type in commands. By default, if a user account is not specified, the `su` command will open a new shell as the root user, which provides administrative privileges.

#### Follow Along

Utilizing the login shell option is recommended, as the login shell fully configures the new shell with the settings of the new user. This option can be specified one of three ways:

```
su -  
su -l  
su --login
```

>\_ Ubuntu PC ▾

Toggle onscreen keyboard

After executing the `su` command, a password is required. On our virtual machines, the password for both the `root` and `sysadmin` accounts is `netlab123`. If you ever forget, it is displayed every time a new virtual machine is started. As a security measure, the password will not be visible as it is typed.

```
sysadmin@localhost:~$ su -  
Password:  
root@localhost:~#
```

Note the command prompt has changed to reflect that you are now logged in as the `root` user. To logout and return to the `sysadmin` account, use the `exit` command. Note the prompt changes back:

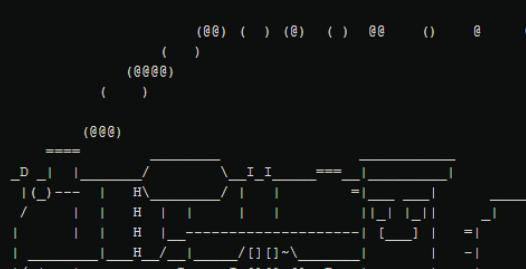
```
root@localhost:~# exit  
logout  
sysadmin@localhost:~$
```

To avoid executing any sensitive commands, we've configured the Steam Locomotive command, the `sl` command, to require administrative access. If the command is executed as `sysadmin`, it fails:

```
sysadmin@localhost:~$ sl  
-bash: /usr/bin/sl: Permission denied
```

Use the `su` command to switch to the root account and execute the `sl` command with administrative access:

```
sysadmin@localhost:~$ su -  
Password:  
root@localhost:~# sl
```





Use the `exit` command again to return to the `sysadmin` account.

```
root@localhost:~# exit
logout
sysadmin@localhost:~$
```

## The `sudo` Command

`sudo [OPTIONS] COMMAND`

The `sudo` command allows a user to execute a command as another user without creating a new shell. Instead, to execute a command with administrative privileges, use it as an argument to the `sudo` command. Like the `su` command, the `sudo` command assumes by default the `root` user account should be used to execute commands.

### Consider This

The `sudo` command can be used to switch to other user accounts as well. To specify a different user account use the `-u` option.

Execute the `sl` command as the root user by putting `sudo` in front of it:

### Note

Remember the password is `netlab123`. The prompt for the password will not appear again as long as the user continues to execute `sudo` commands less than five minutes apart.

```
sysadmin@localhost:~$ sudo sl
[sudo] password for sysadmin:
```



Once the command has completed, notice the prompt has *not* changed, you are still logged in as `sysadmin`. The `sudo` command only provides administrative access for the execution of the specified command. This is an advantage as it reduces the risk that a user accidentally executes a command as root. The intention to execute a command is clear; the command is executed as root if prefixed with the `sudo` command. Otherwise, the command is executed as a regular user.

◀ Previous

Next 