Access Controls and Managing Services

Cybersecurity

4.3 Managing Permissions and Services

Class Objectives

By the end of class, you will be able to:

Inspect and set file permissions for sensitive files on the system.

Manage and monitor services on the system, and remove unused services.

Create and assign users for services.

Access Controls

Like Google Docs, Linux has **access controls**, which grant permission to access documents and files on a host.

Managing Access Controls in Linux

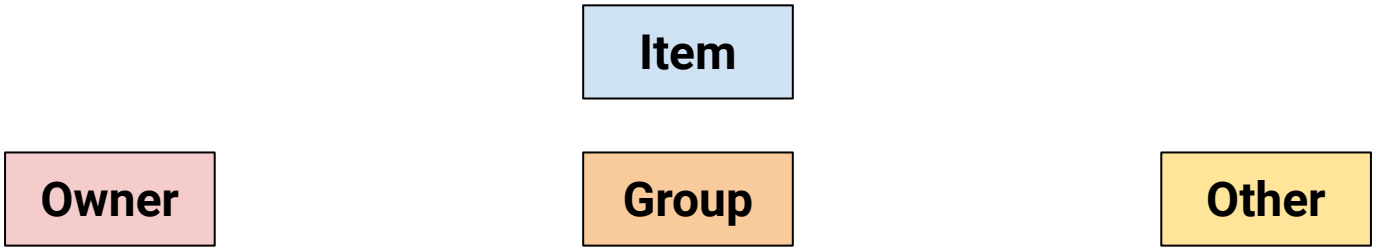
Linux treats resources, programs, and files as **items**.

Item

Managing Access Controls in Linux

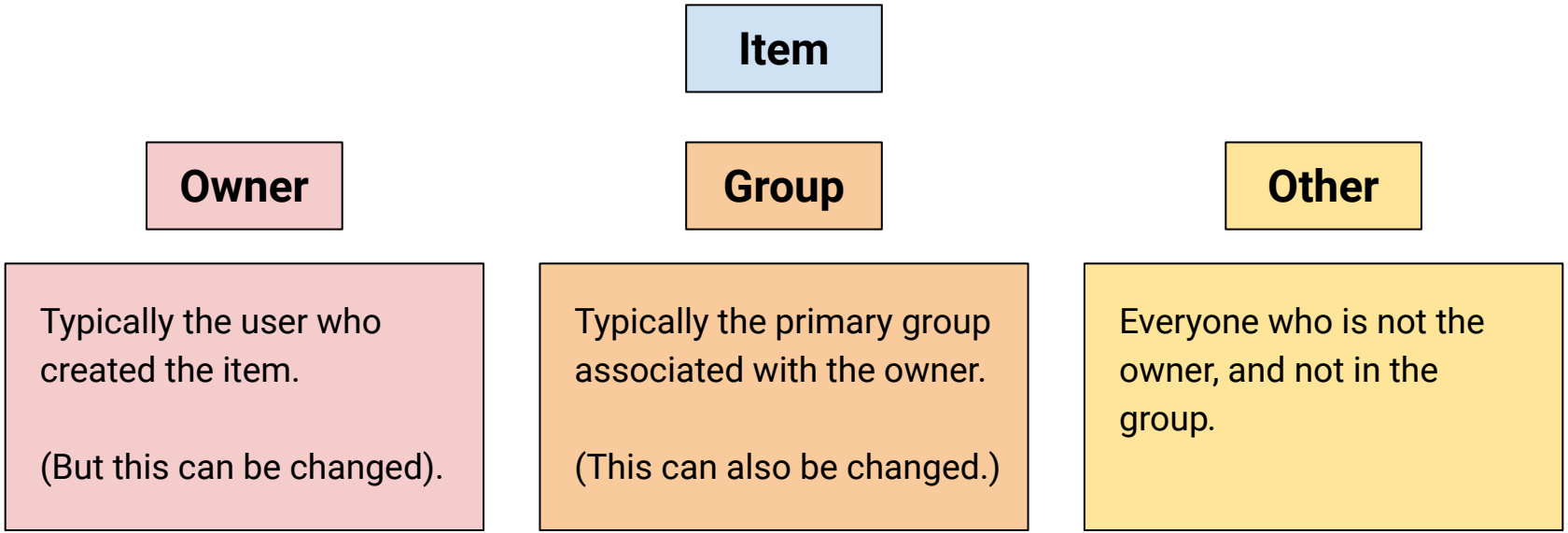
Managing access controls in Linux

Linux manages access controls for the item, the *group* associated with the item, and *others*.



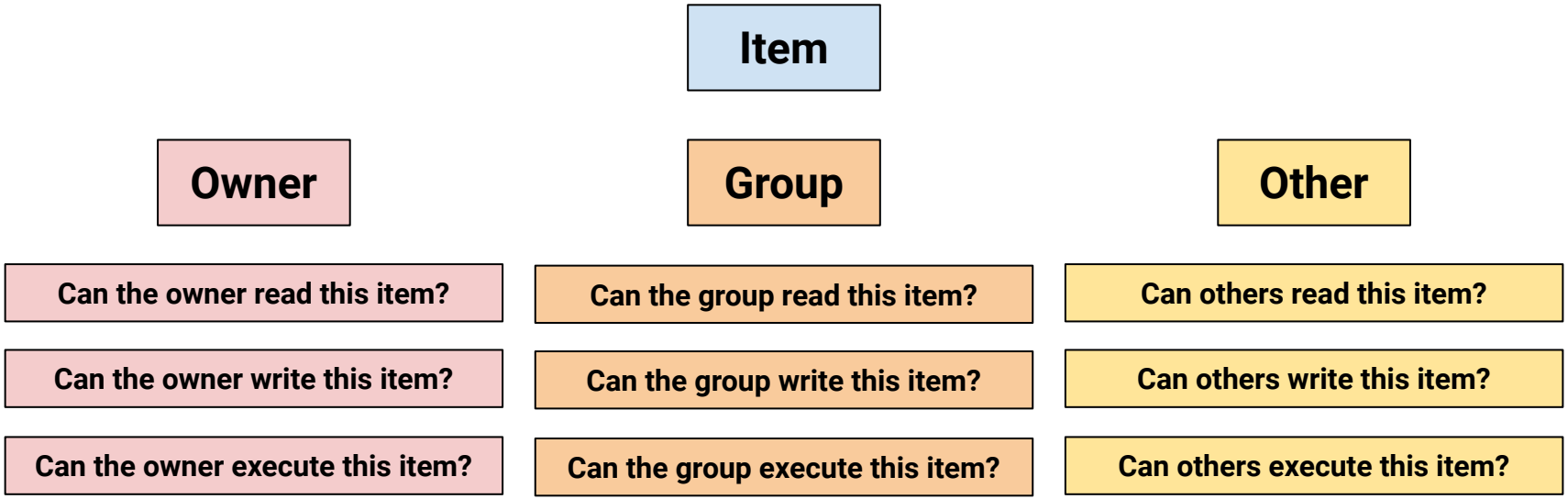
Managing Access Controls in Linux

Managing access controls in Linux is all about the *owner* of the item, the *group* associated with the item, and *others*.



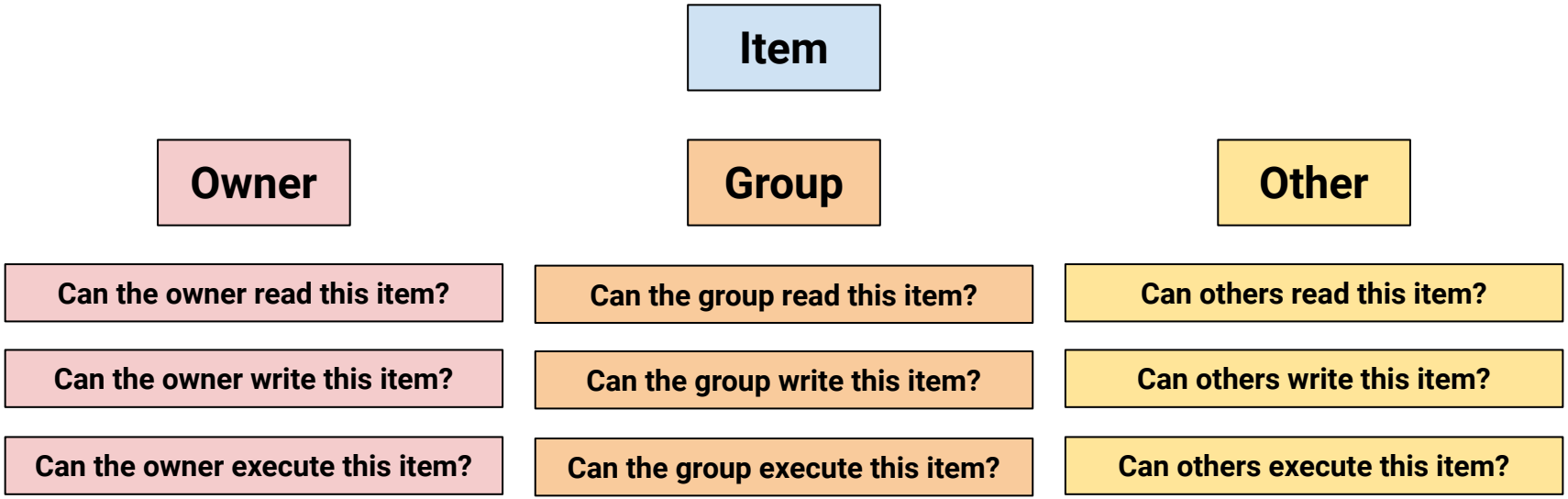
Managing Access Controls in Linux

For each of these categories, there are three actions that we can allow or prevent: **read, write, execute.**



Managing Access Controls in Linux

Assigning these permissions is called **Discretionary Access Control (DAC).**
It is *discretionary* because permissions can pass from one item to another.

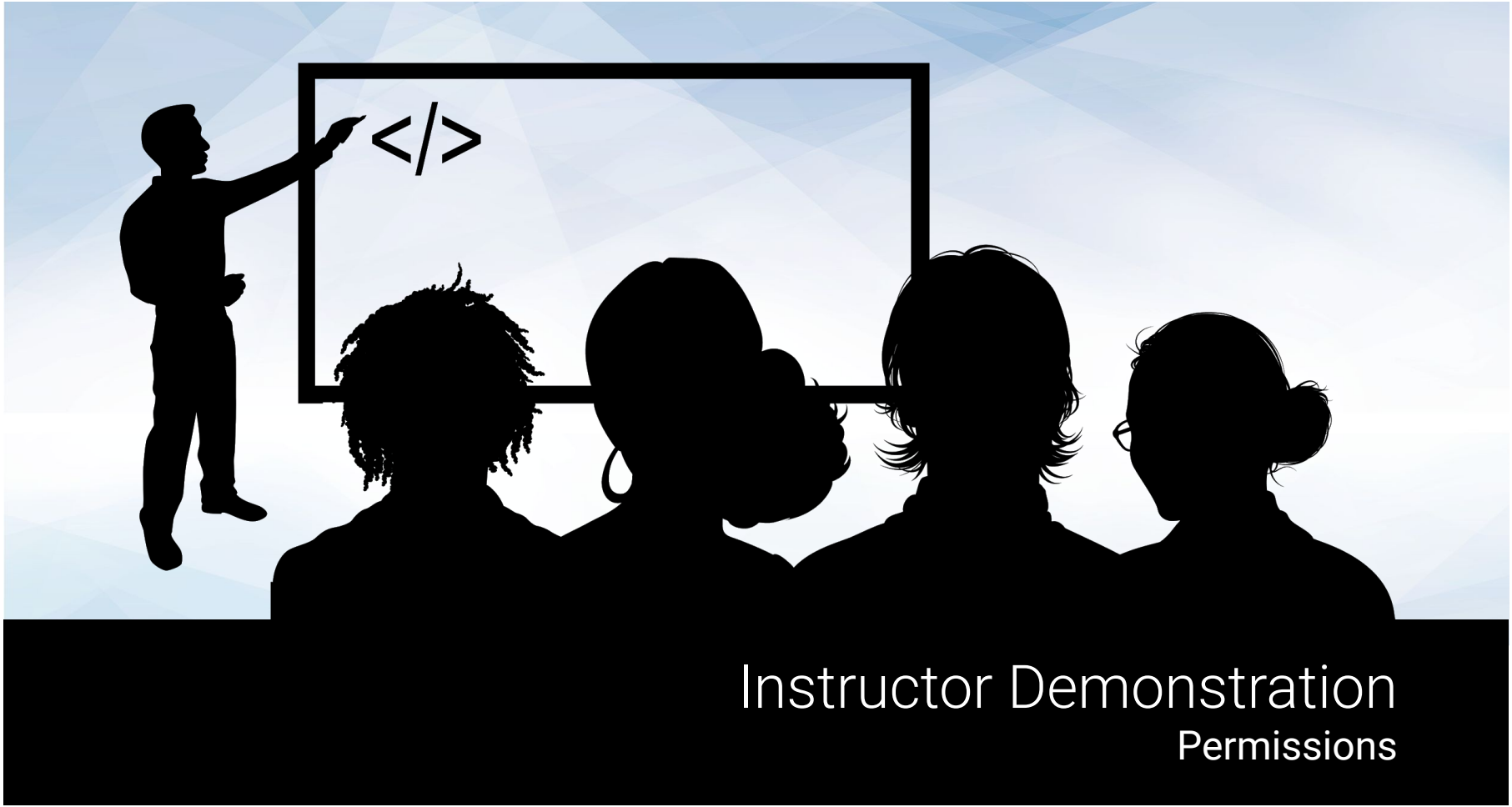


Permissions Demo

In the upcoming demo, we'll create a file and a directory, observing default permissions. Then, we will change the permissions to deny certain groups and users access.

To read and manipulate these file permissions, we'll use these commands:

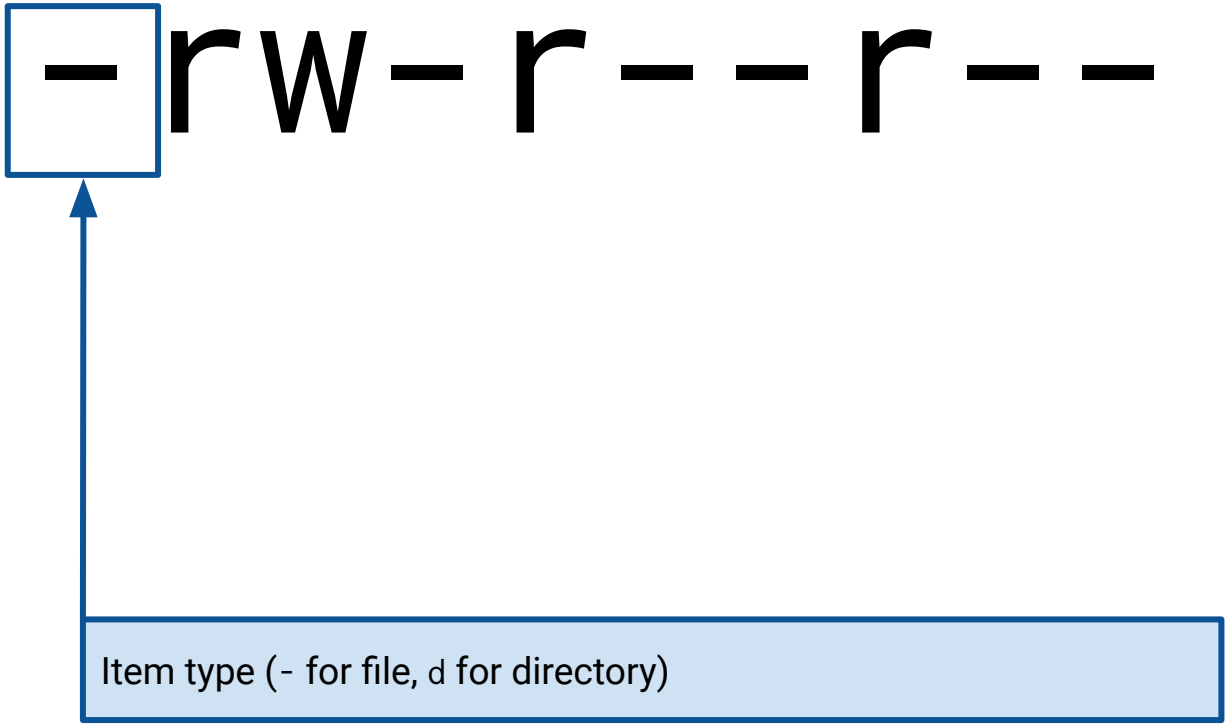
`ls -l`	Show the permissions info.
`chmod`	Change the permissions info.
`chown`	Change the owner and group of a file.



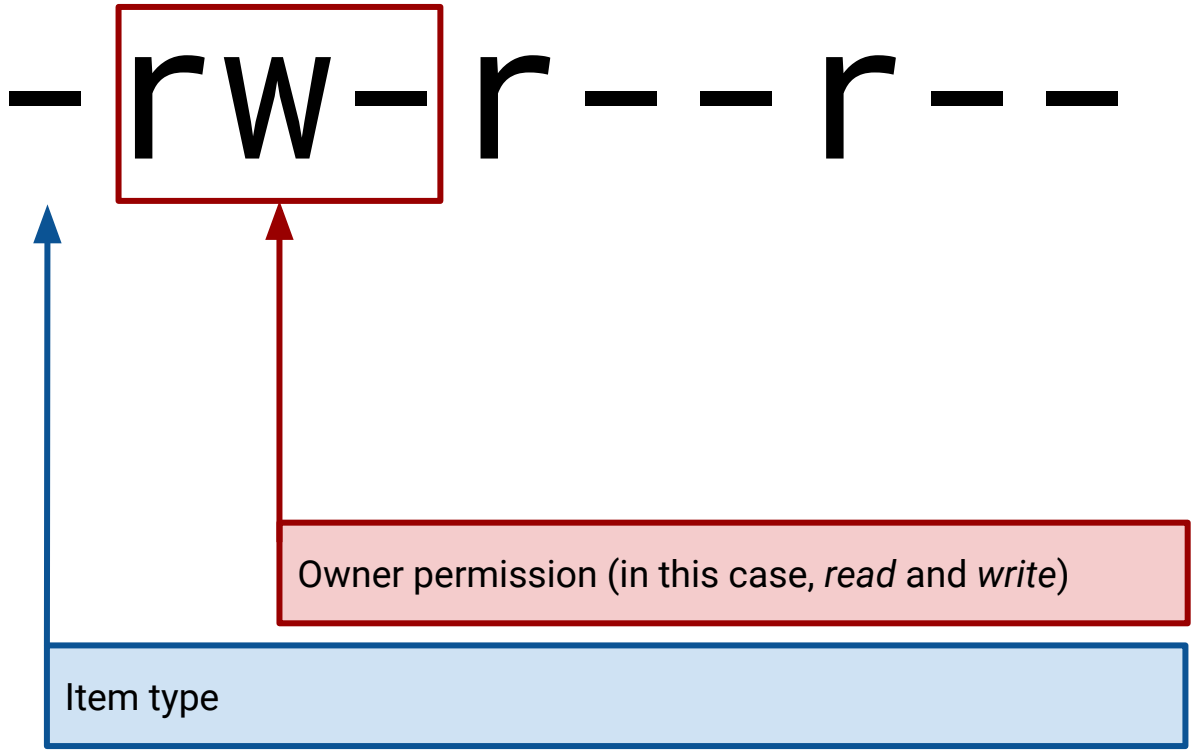
Instructor Demonstration

Permissions

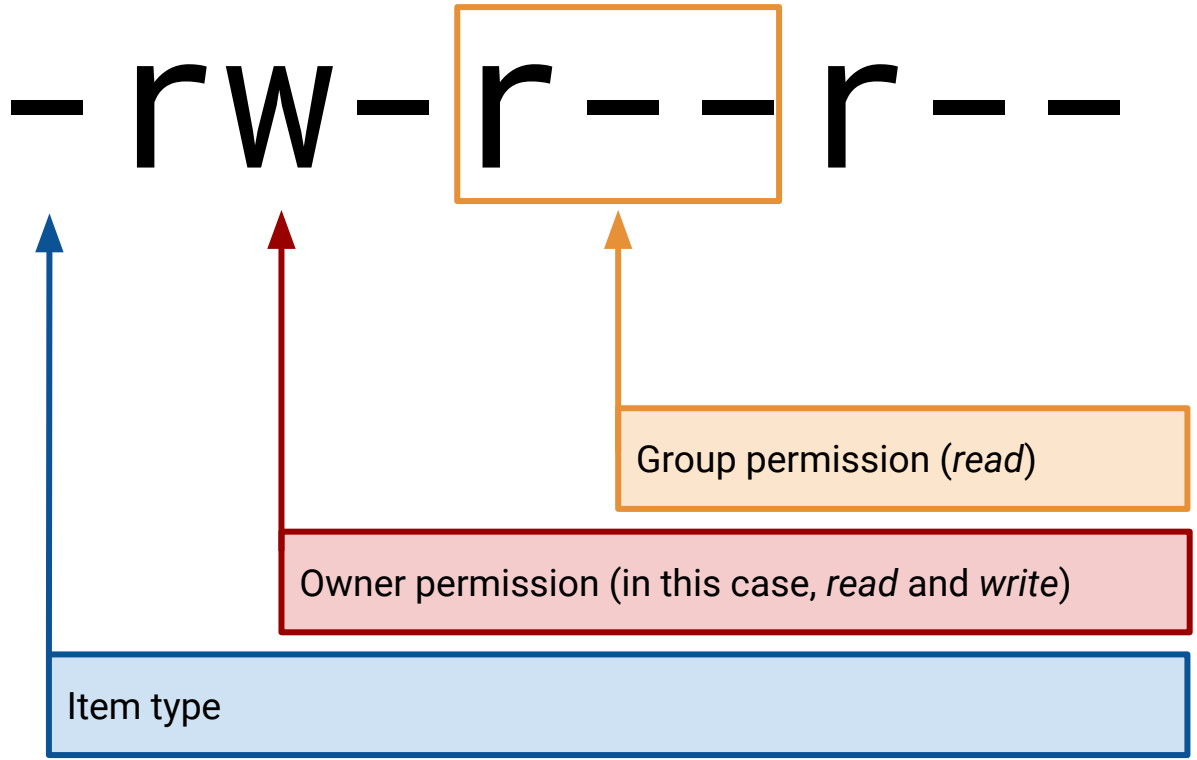
Inspecting File Permissions



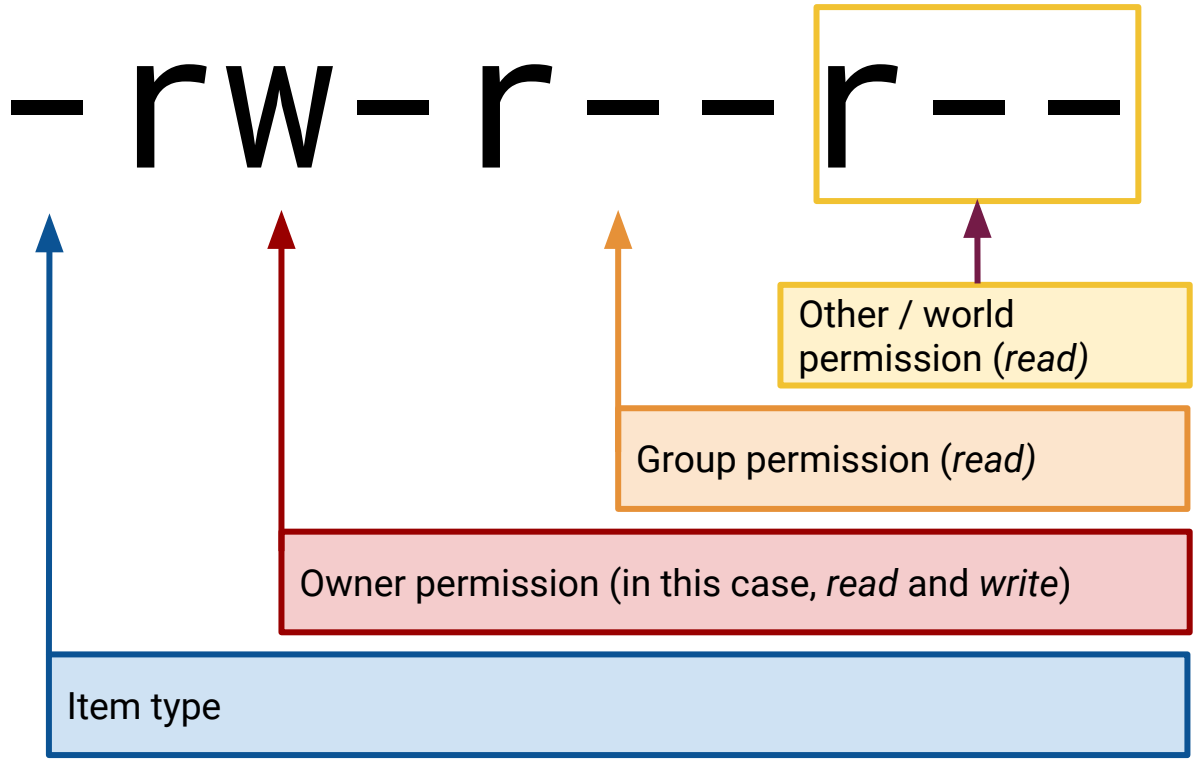
Inspecting File Permissions



Inspecting File Permissions



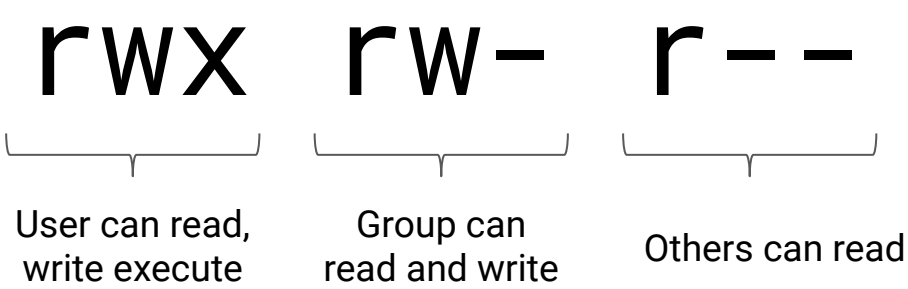
Inspecting File Permissions



Changing File Permissions

File permissions can be set using two different notations: **symbolic** and octal.

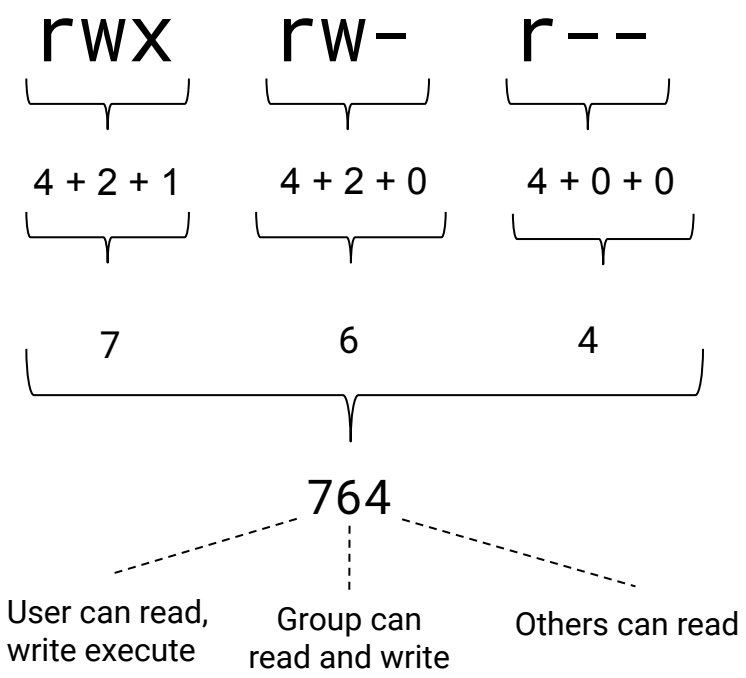
Symbolic Notation	
`r`	read
`w`	write
`x`	execute

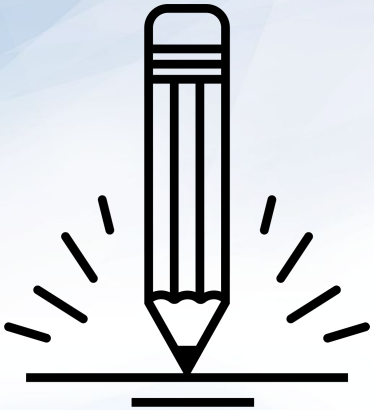


Changing File Permissions

File permissions can be set using two different notations: symbolic and **octal**.

Octal Notation				
	4	2	1	
0	-	-	-	No permission
1	-	-	x	Only execute
2	-	w	-	Only write
3	-	w	x	Write and execute
4	r	-	-	Only read
5	r	-	x	Read and execute
6	r	w	-	Read and write
7	r	w	x	Read, write, and execute





Activity: Access Controls and Permissions

In this activity, you will inspect and set file permissions on a few of the most sensitive items on a Linux system.

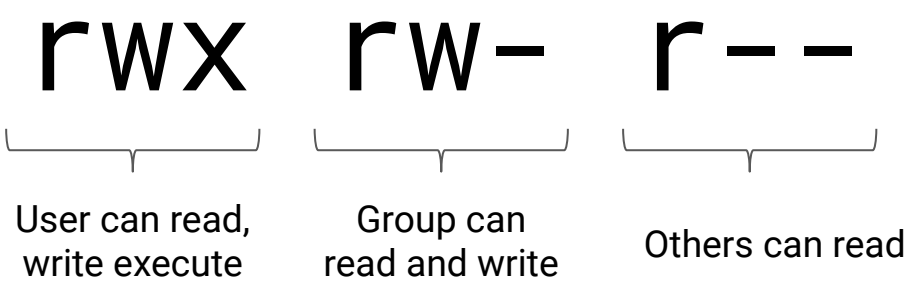
Suggested Time:
25 Minutes

Times Up! Let’s Review.

Recap: Permissions

How permissions apply to each specific file and folder with `r`, `w`, and `x`.

Symbolic Notation	
`r`	read
`w`	write
`x`	execute



Permissions

How to view and apply permissions to an item's user, group, and other.

Users

Every file and program on a Linux system has permissions.

These permissions tell the system which users can access the file or run the program.

Groups

Users can be placed in groups, which can have special permissions that apply to all members of the group.

Root

File and program permissions apply to all users in a system, *except* the root user.

The root user (or super user) has complete access and can perform any action.

Permissions

We can use **sudo** user to invoke the **root** user and bypass any permissions.

`ls -l`	To show the permissions info.
`chmod`	To change the permissions info.
`chown`	To change the owner and group of a file.

Permissions

We can assign **sudo** for a specific command for a specific user.

`whoami`	To determine the current user.
`su`	To switch to another user, in this case the root user.
`sudo`	To invoke the root user for one command only.
`sudo -l`	To list the `sudo` privileges for a user.
`visudo`	To edit the `sudoers` file.


Break

Managing Services


Servers are computers that offer services to other computers.

Managing Services

A service is a function or capability that a machine makes available to another.

For example, file sharing services allow computers to send and receive data.

Managing Services

Some services, like Tripwire, are only run locally on the server and are not provided to other computers. These services are packages that can be installed and removed just like other programs.

Services and Security

Services and Security

Attackers can manipulate services into doing things that they are not designed to do.

Services and Security

For example: Samba (SMB), the file sharing protocol, allows users to view, download, and store files remotely.

Finding and Stopping SMB Demo

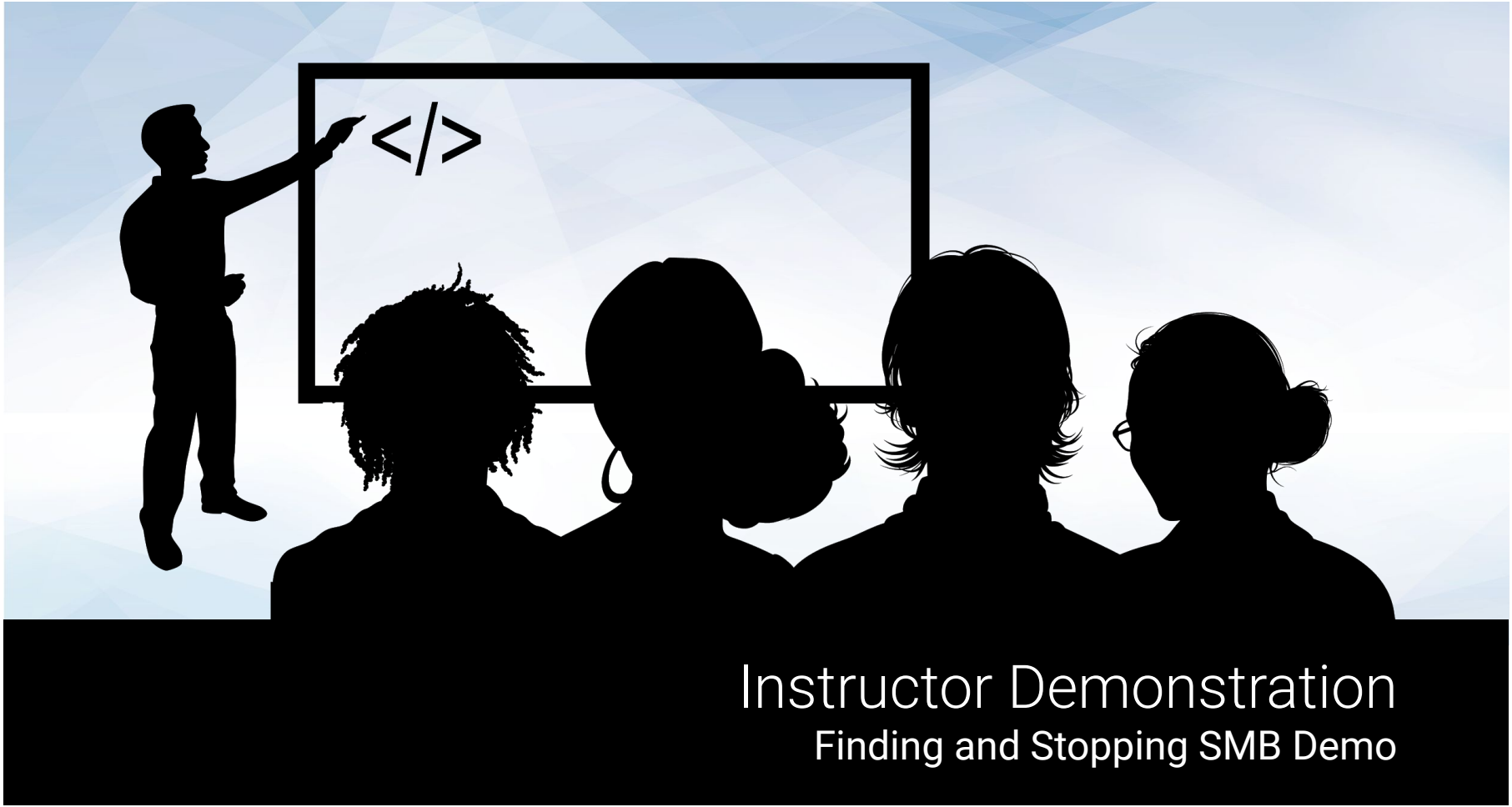
If a malicious user is able to gain access to a shared folder, they can exfiltrate, alter, or delete sensitive files.

- In this example, the server has already been compromised.
- In the following demo, we will stop the SMB service, and then uninstall it from the system.

Finding and Stopping SMB Demo

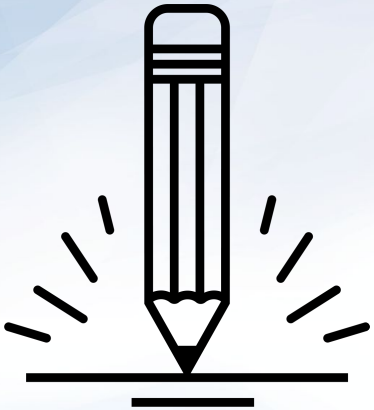
This will require the following steps:

- Listing all running services.
- Identifying the Samba service in the list to confirm it's running, then stopping it.
- Ensuring Samba doesn't start when the machine is started up.
- Ensuring Samba is no longer running.
- Uninstalling the Samba service completely.



Instructor Demonstration

Finding and Stopping SMB Demo



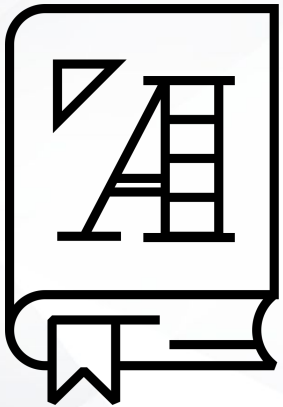
Activity: Managing Services

Your senior administrator wants you to audit the services being run by the server and shut down old and unused services.

Suggested Time:
25 minutes

Time’s Up! Let’s Review.

Service Users

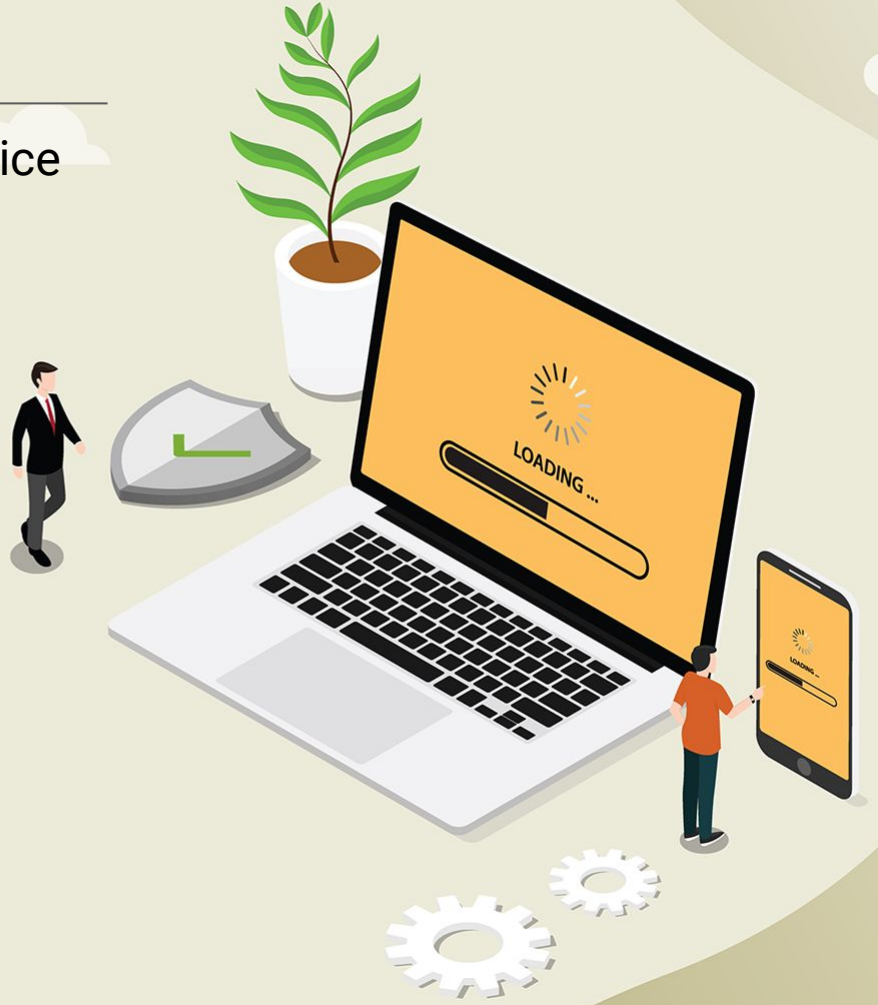


Some services are not run by real users. They are run by specific **service users** that are dedicated to running their own specific service.

Service Users

Typically, when you install a service with the package manager, a service user is automatically created and configured.

Running services under a dedicated user offers several security benefits. It makes it easier to start, stop, and manage the service, and control which files the service needs to access.



A service user usually has a system **UID less than 1000** and cannot log in to use a shell.

Service Users

Since service users aren't humans who need to log into and interact with the machine, it's best practice to ensure that users cannot log into an interactive shell using a service username.

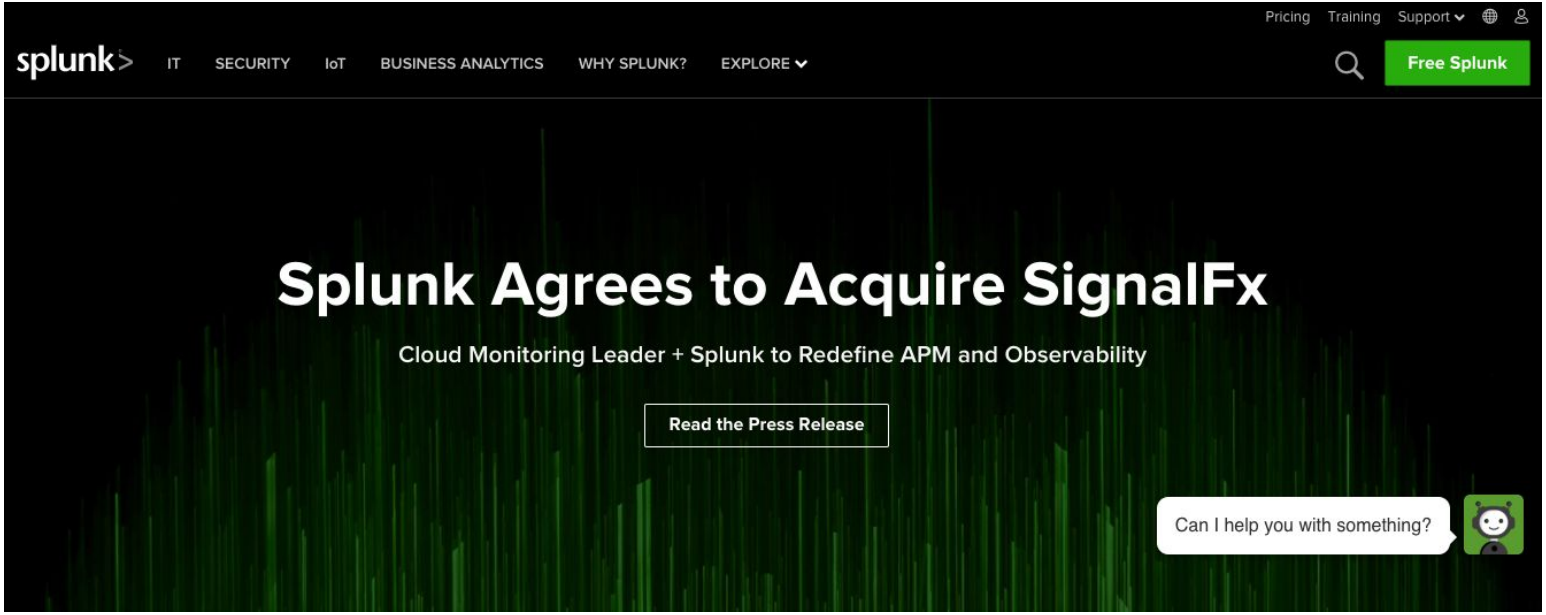
Scenario: Setting Up and Adding Service Users Demo

Your senior administrator asked you to follow up on your uninstallation of unused services. You must now ensure the services' corresponding users have also been removed from the system.

Previously, you disabled vsftpd, but its service user, ftp, still exists

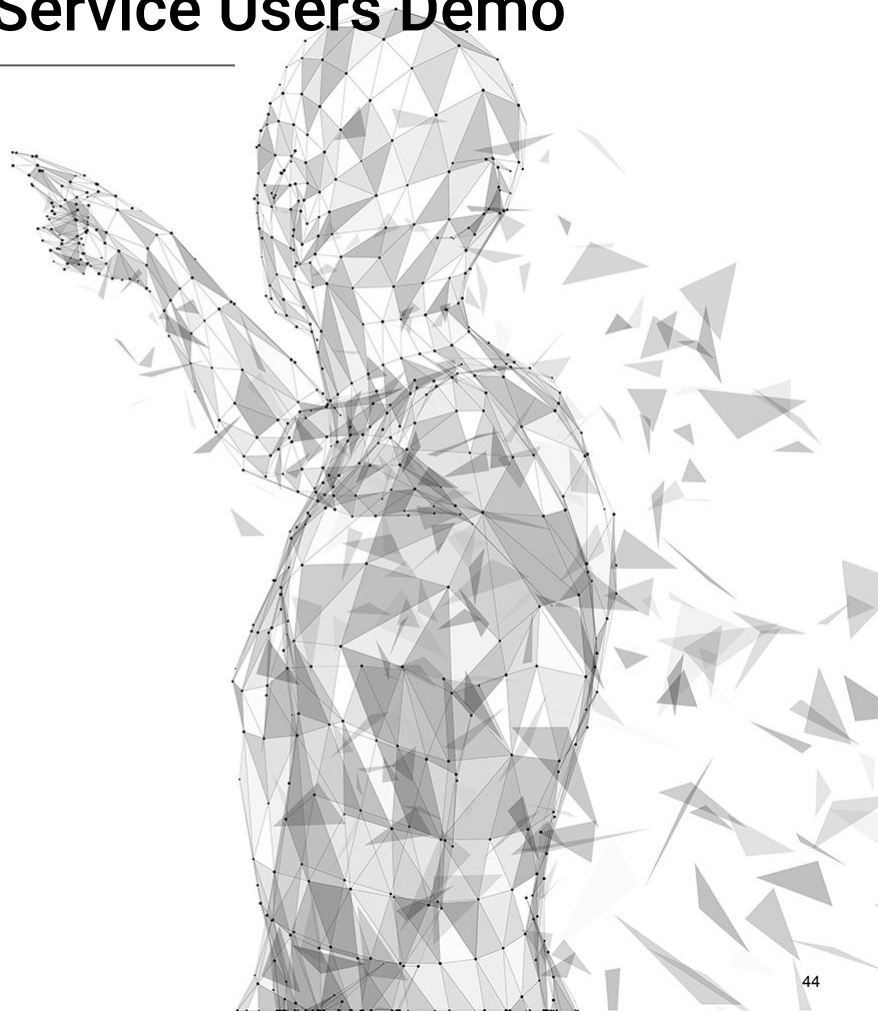
Scenario: Setting Up and Adding Service Users Demo

Your senior administrator also plans to install a security service called Splunk to collect and analyze logs for suspicious activity. Like Tripwire, Splunk makes it easier for admins and security personnel to detect and stop malicious behavior.



Scenario: Setting Up and Adding Service Users Demo

Your senior administrator told you that they'll handle the installation and configuration themselves, but have requested that you create a service user that they can use later.



Scenario: Setting Up and Adding Service Users Demo

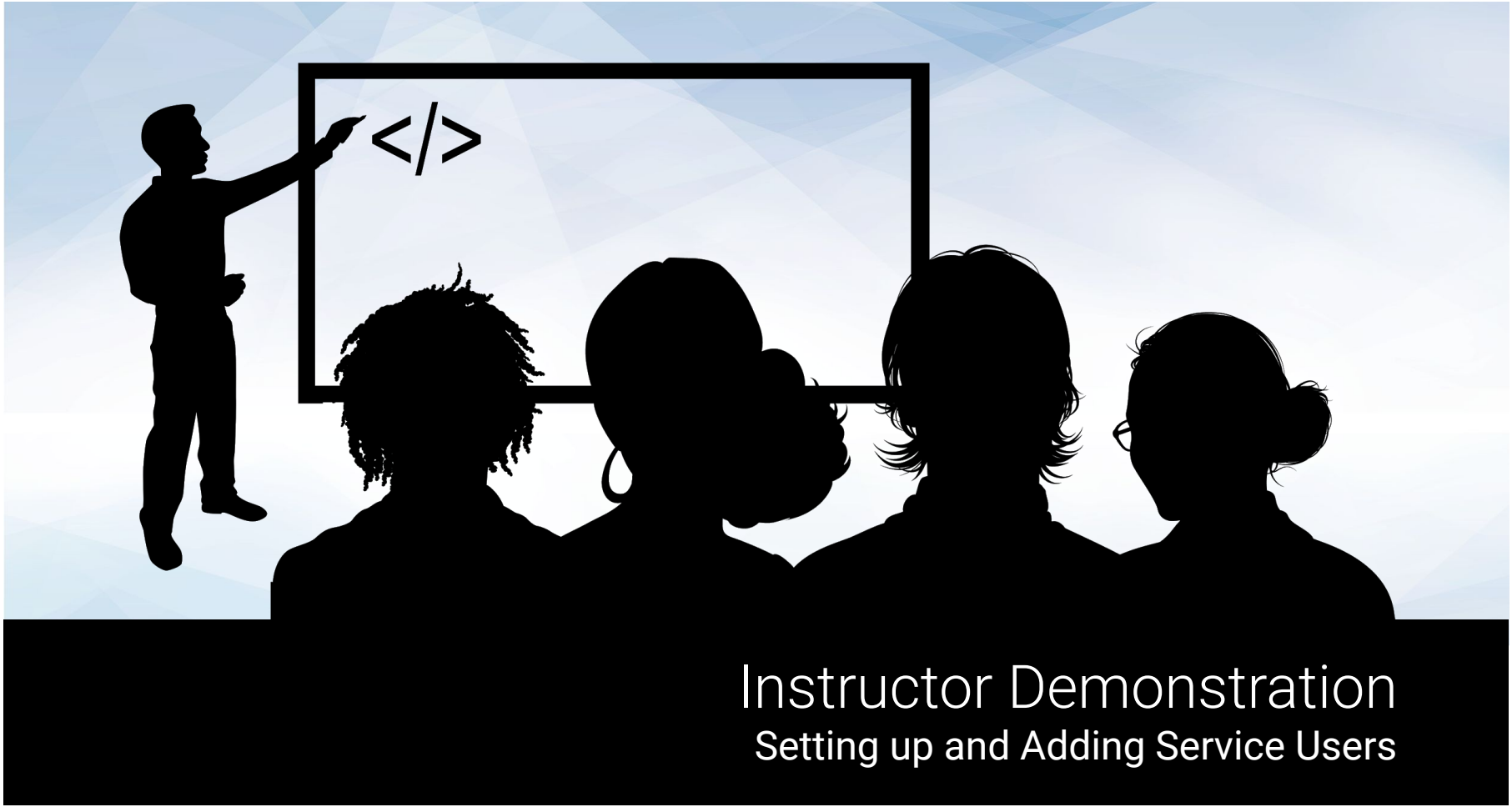
Completing this task will require the following steps:

01 **Delete**

- Deleting an old, unused service user with **`deluser/`**.

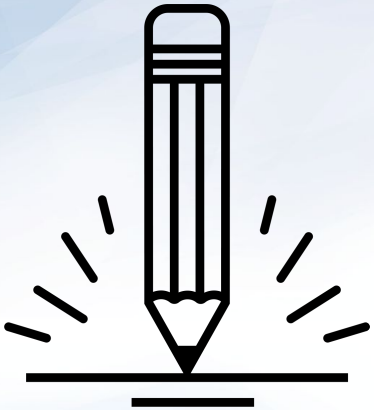
02 **Create**

- Creating and validating a new service user with **`adduser`**.



Instructor Demonstration

Setting up and Adding Service Users



Activity: Service Users

Your senior administrator would like you to remove any old service users from the system and create a new user dedicated to running Tripwire.

- Use **`adduser`** and **`deluser`** with the correct flags to clean up the system and create this new Tripwire user.
- Tripwire can only be run as **`root`** , so you must add a line to the **`sudoers`** file to allow this.

Suggested Time:
25 minutes

Time’s Up! Let’s Review.

Homework

In this week's homework, you will practice all the hardening steps we learned this week, this time on a new system.

You will also run a few new tools: **`chkrootkit`** and **`lynis`**.

Questions?


The End