



# Access Controls





Cybersecurity  
4.2: Linux Access Controls



# Class Objectives

---

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

-  Audit passwords using `john`.
-  Elevate privileges with `sudo` and `su`.
-  Create and manage users and groups.
-  Inspect and set file permissions for sensitive files on the system.

# A Brief Introduction to Hashes and Password Cracking

# Password Hashes

---

A hash is a cryptographic function that takes data as input and translates it to a string of different, seemingly random data.

## My Plain Text Password

ApPles20rang3s93

## My Password Hash

579de3a38386c62a1eca4600e3882b8b

A hash will always output the same string for the same input data.

# Password Hashes

The hash is stored in the shadow file.

When a user logs in, the hash of the submitted password is compared to the hash stored in `etc/shadow`.

-----  
If the hashes match, the user's logged in.



# Password Cracking

---

A hash is a cryptographic function that takes data as input and translates it to a string of different, seemingly random data.

## My Plain Text Password

ApPles20rang3s93

## My Password Hash

579de3a38386c62a1eca4600e3882b8b

Password cracking tools **cannot** reverse a password hash.

# Password Hashes

---

Password cracking tools **cannot** reverse a password hash.

Instead, they use a wordlist of potential passwords and create hashes for each one.

-----

This form of password hacking is called a **brute force attack**.





The more random and  
lengthy the password, the  
longer it will take to crack.



# How secure is my password?

---

Go to [howsecureismypassword.net](https://howsecureismypassword.net).

HOW SECURE IS MY PASSWORD?

ENTER PASSWORD

Sponsored by [Dashlane](#): never forget another password

# Secure Password Takeaways

---

01

If a system requires passwords of only sixteen characters, the password will be relatively strong, even if it contains words.

02

Add a few extra characters and it will become exponentially more secure.

03

In contrast, if using all random characters, a password still must be at least 10 characters to be very effective.

# Cracking Passwords

---

Modern password cracking software works using the following steps:

01

Takes a list of hashes as input.

02

Hashes passwords from a given password list and compares each hash to the list of hashes it was given.

03

If it matches a hash, it gives outputs of what password was used to create the hash.

# Password Cracking and John the Ripper

---

Modern password cracking software works uses the following steps:

1. Takes a list of hashes as input.
2. Hashes passwords from a given password list and compares each hash to the list of hashes it was given.
3. If it matches a hash, it gives outputs of what password was used to create the hash.

John the Ripper is a popular modern software that uses this technique to crack a wide variety of hashes.





# Instructor Demonstration

John the Ripper



## **Activity:** Let's Talk to John

In this activity, you will continue your role as a junior administrator auditing a system. Now, the focus is on passwords.

- You will use john to crack the password hashes for all of the users on our system.

**Suggested Time:**  
20 Minutes





**Times Up!** Let's Review.



Privileges, root, sudo, and su





**We've used `sudo` for several commands in this unit.  
Why do you think some commands require  
`sudo` and others don't?**

# Privileges and Users

---

Why do we need to use sudo for some commands and not others?

## Users

Every file and program on a Linux system has permissions.

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

## Groups

Users can be placed in groups, which can have their own permissions.

## Root

File and program permissions apply to all users *except* the root.

The root user (or super user) has complete access to the system and can perform any task.

# Root Access

---

When attackers try to gain access to a system, they often try to gain **root access**.



Secure Linux systems do not allow just anyone to log in as the root user on the system.



`sudo` (superuser do) can grant a user root privileges for one command.



When the one `sudo` command is done, users are reverted to their usual access.



`sudo` can also control which commands the user can run as root.

# Su Demo

---

In the following demo, we will attempt to update all of our existing software packages:

- If our privileges do not allow us to do so, we will first use `su` to switch directly to the root user.
- We'll show the dangers of working directly as the root user.
- Then, we'll do the same updates by using `sudo` instead and show why this is the more secure option.



# Instructor Demonstration

su vs. sudo

# Sudo Demo Summary

---

In the previous demonstration, we covered the following commands:

<code>whoami</code>	Determines the current user.
<code>su</code>	Switches to another user, in this case the root user.
<code>sudo</code>	Invokes the root user for one command only.
<code>sudo -l</code>	Lists the <code>sudo</code> privileges for a user.
<code>visudo</code>	Edits the <code>sudoers</code> file.



## Activity: Sudo Wrestling

In this activity, you will continue your role as a junior administrator auditing the system:

- The senior administrator has asked you to audit the system for sudo and root access, making sure no users other than the admin user have access to any sudo use.
- You must log in as each user, check their privileges, edit the sudoers file, and look for anything else suspicious.

**Suggested Time:**  
25 Minutes





**Times Up!** Let's Review.



A close-up, high-angle shot of a computer keyboard. The central focus is a large, white, rectangular key with rounded corners. On this key, there is a dark blue icon of a coffee cup with three wavy lines above it representing steam. Below the icon, the word "Break" is printed in a dark blue, serif font. The key is set against a light-colored, textured keyboard surface. Surrounding the main key are other keys, including one with a double quote symbol to the left and one with a dash/slash symbol to the right, all slightly out of focus.

Break

# Users and Groups

# Users and Groups

Users on a Linux system can be added to groups.

Linux has the ability to create groups of users for functions like file and service sharing.

If a company has different departments, like Sales, Accounting, and Marketing, a Linux admin can create a group for each department. **Only users in each group can access files owned by the group.**



# Users and Groups

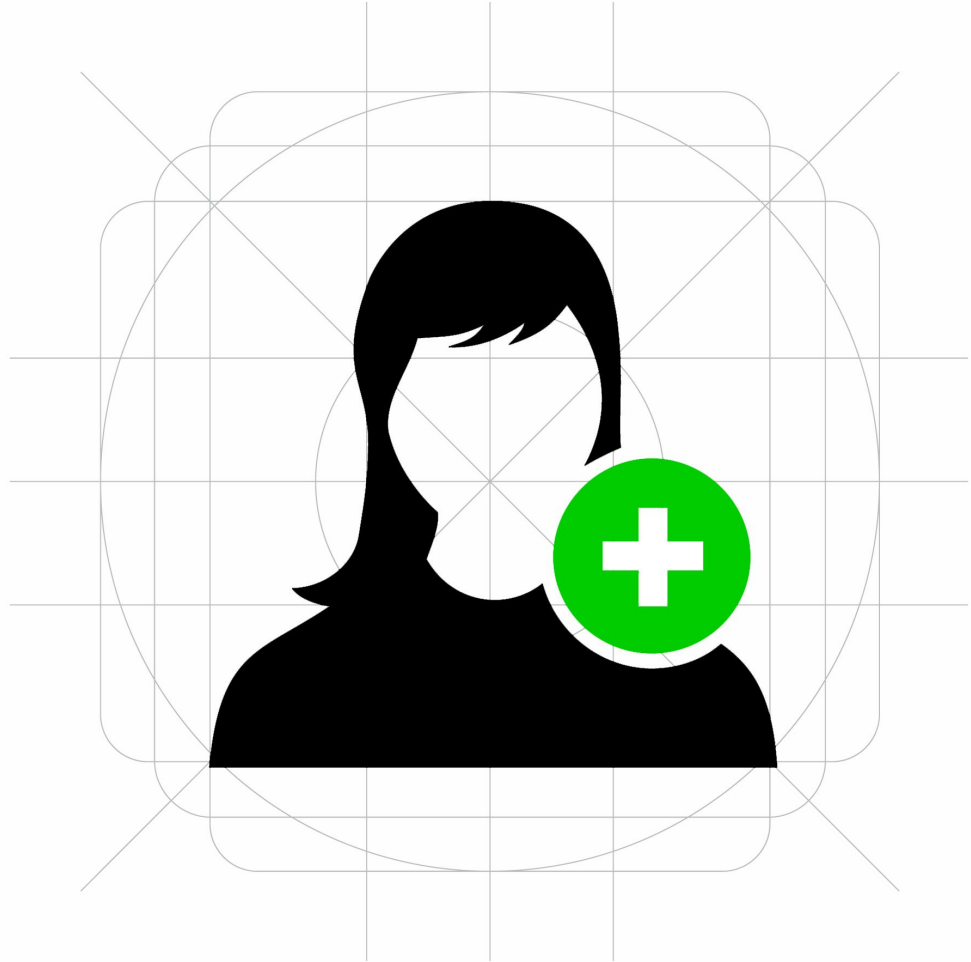
---

A system admin must know how to to add and remove users to a system, add and remove groups, and add or remove users from groups.

Soon we'll discuss commands specifically used for user and group management.



**But first, let's see how Linux identifies users and groups in the system.**



# id command

---

Linux identifies users and groups in the system using the **id** command:



Linux associates a specific number with each user, known as the **user ID (UID)**.



When Linux needs to identify a user, it uses the UID, not the username.



System users have a UID that is **less than 1000**.



Standard users have a UID that is **greater than 1000**.



The root user always has the UID of **0**.

# Users and Groups Demo Scenario

---

In the upcoming demo, we'll dive into more actions for user and group management using the following scenario:

**Your company recently made changes to the developer team.**



Mike, a lead developer,  
has left the company.

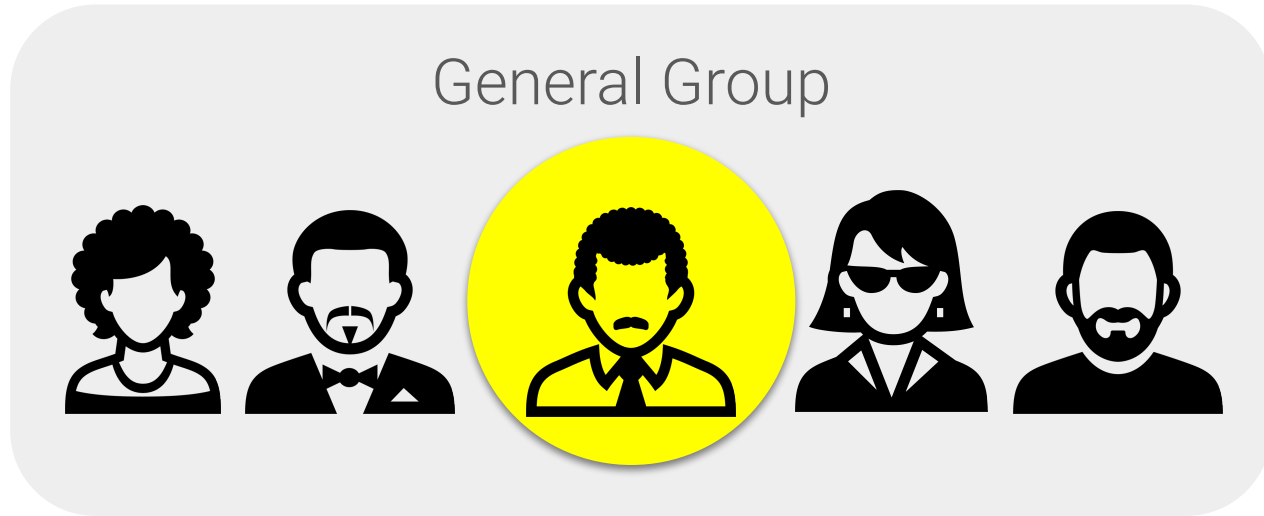


Joseph has joined as a  
new junior developer.

# Users and Groups Demo Scenario

---

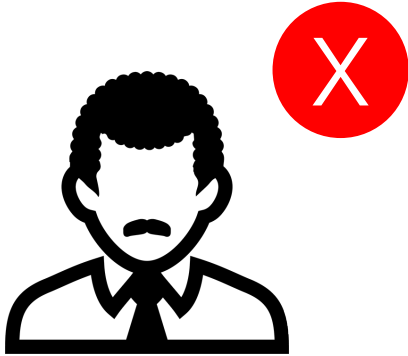
The company's Linux system has never been set up properly with a developers group. Instead, **Mike was part of the general group.**



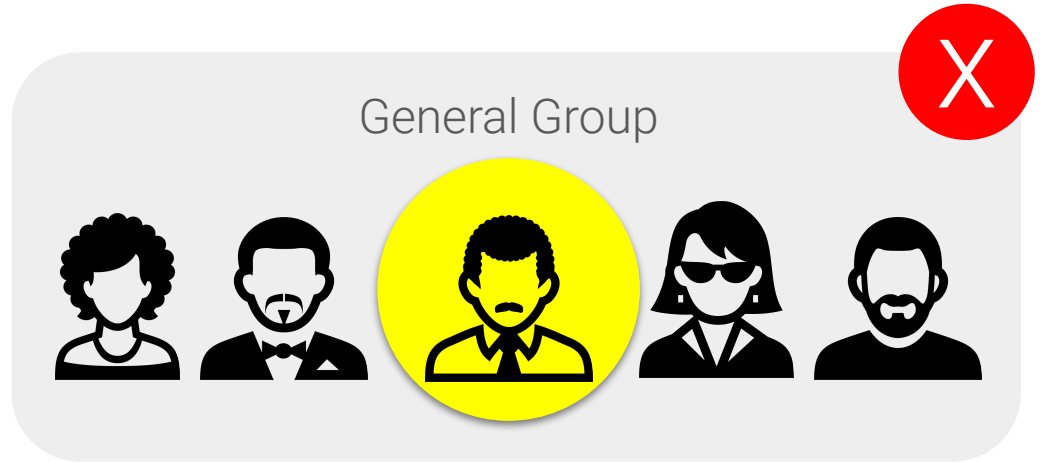
# Users and Groups Demo Scenario

---

As the sysadmin for this system, you need to remove Mike from the general group, remove the general group, and delete Mike from the system.



Mike, a lead developer, has left the company.





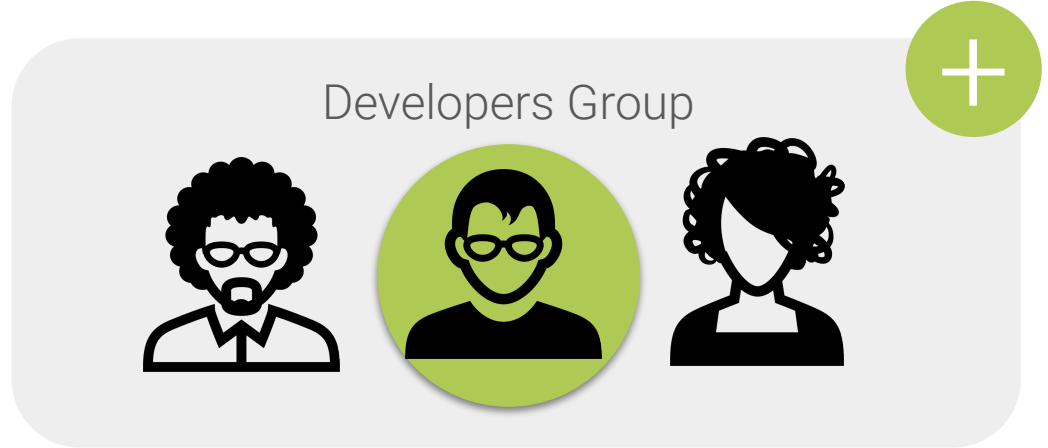
# Users and Groups Demo Scenario

---

Then, you need to add Joseph to the system, create a developers group, and add Joseph to this group.







Joseph has joined as a new junior developer.



# Users and Groups Scenario

To accomplish these tasks, we will use the following commands:

<code>groups</code>	Get group info for the user <code>mike</code> .
<code>usermod</code>	Lock Mike's account to prevent him from logging in.
<code>usermod</code>	Remove the user <code>mike</code> from the <code>general</code> group.
<code>deluser --remove-home</code>	Delete the user <code>mike</code> .  
<code>delgroup</code>	Delete the <code>general</code> group.
<code>adduser</code>	Create the user <code>joseph</code> .  
<code>addgroup</code>	Create a <code>developer</code> group.
<code>usermod</code>	Add the user <code>joseph</code> to the <code>developer</code> group.



# Instructor Demonstration

## Users and Groups



## **Activity:** Users and Groups

Your senior administrator has asked you to audit all the users and groups on the system.

- You must create a new group for the standard users and remove users from the sudo group.
- In the previous activity, you found some malicious users. Now, you will remove them from the system entirely.

**Suggested Time:**  
20 Minutes





**Times Up!** Let's Review.

Any Questions?

*The  
End*