

**Universidad Autónoma del Estado de México**

**Unidad Académica Profesional Tianguistenco**

**Ingeniería en software**

**Unidad de aprendizaje:**

**Seguridad Informática**

**Profesor:**

Martín García Ávila

**Alumno:**

Andrés Alvir Guzmán

**Fecha de entrega:** 19/02/2023

|  |
| --- |
| **Objetivo** |

Realizar una práctica utilizando los protocolos de AAA de forma personal y utilizando ChatGPT.

|  |
| --- |
| **Desarrollo** |

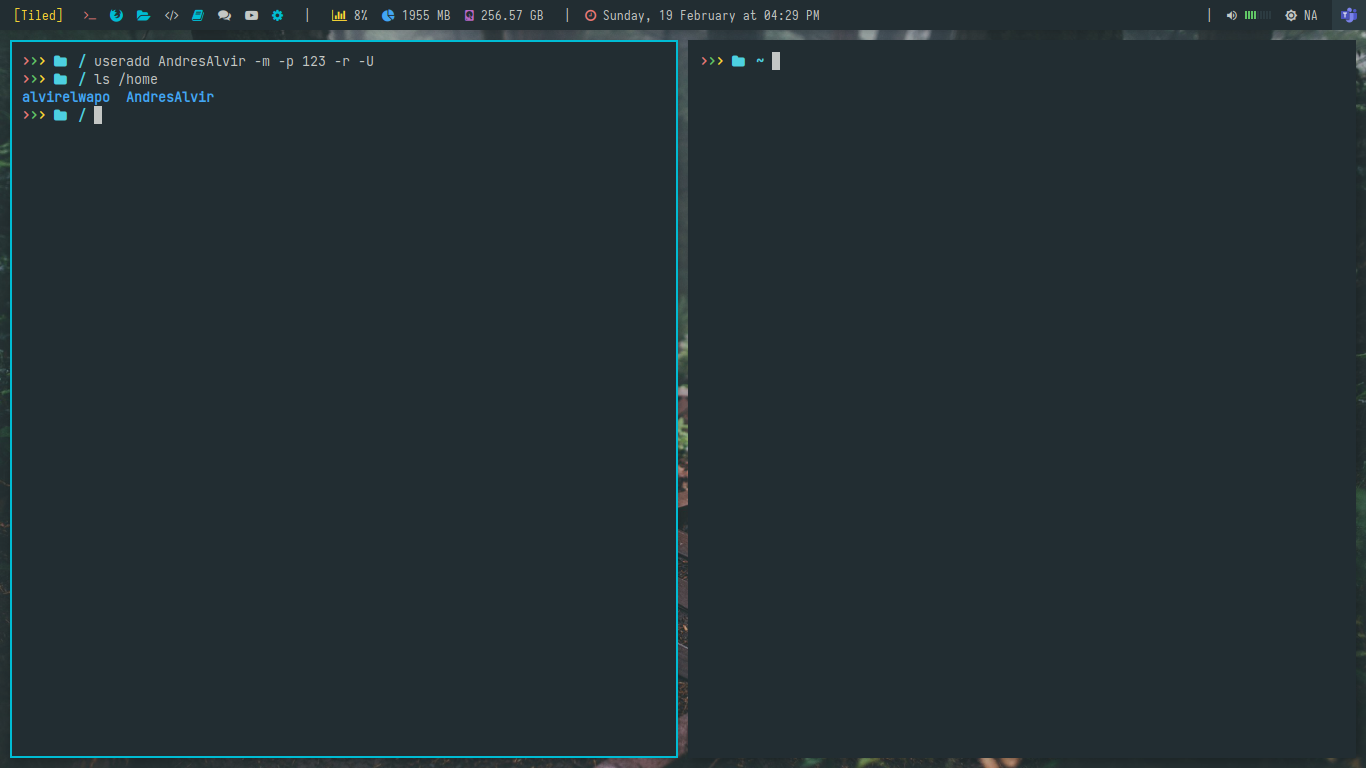
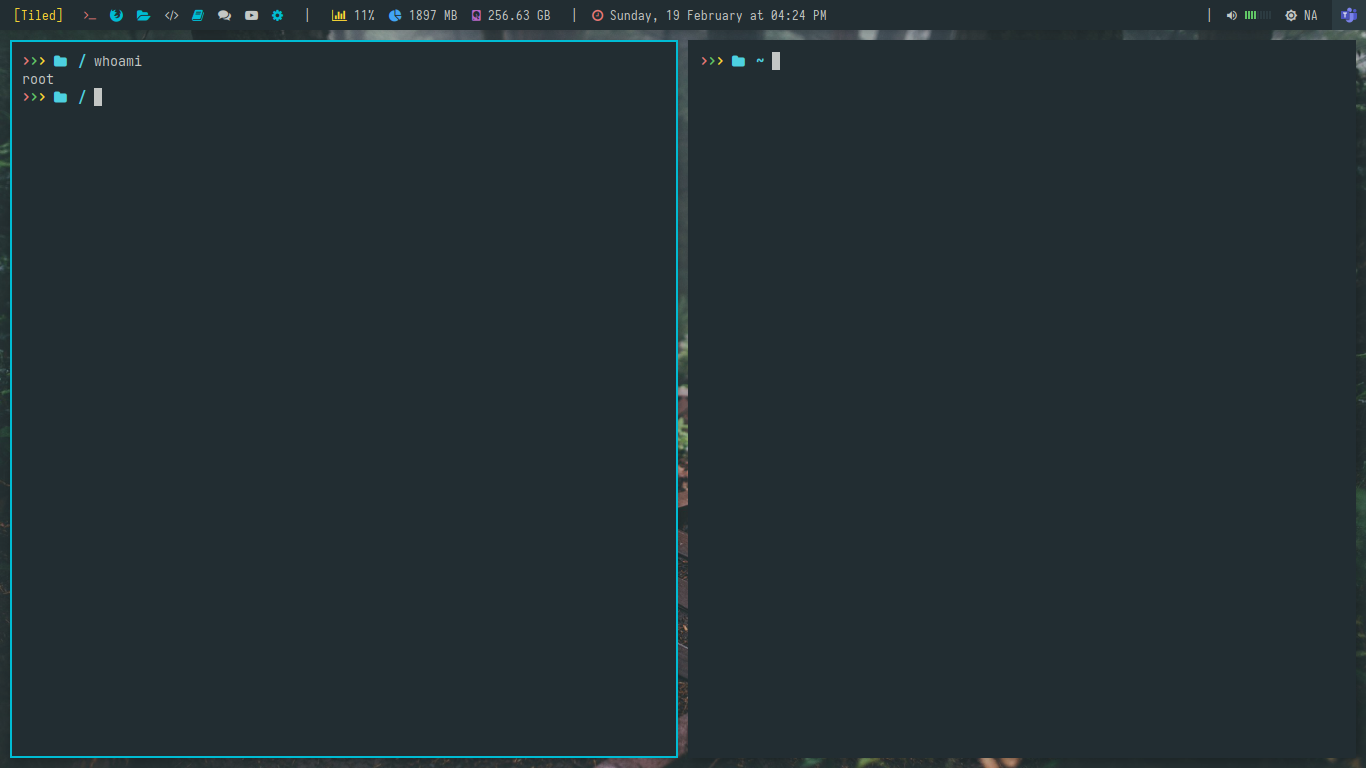
Al principio no comprendí muy bien el punto de la exposición de mi compañero, pensé que éste tipo de protocolos solo existían a través de aplicaciones como Google Authenticator, etc.

Después de razonarlo un rato y comprender a qué se refería Authentication, Authorization and Accounting. Logré deducir que cualquier sistema considerado siquiera remotamente seguro debe tener éstas 3 características.

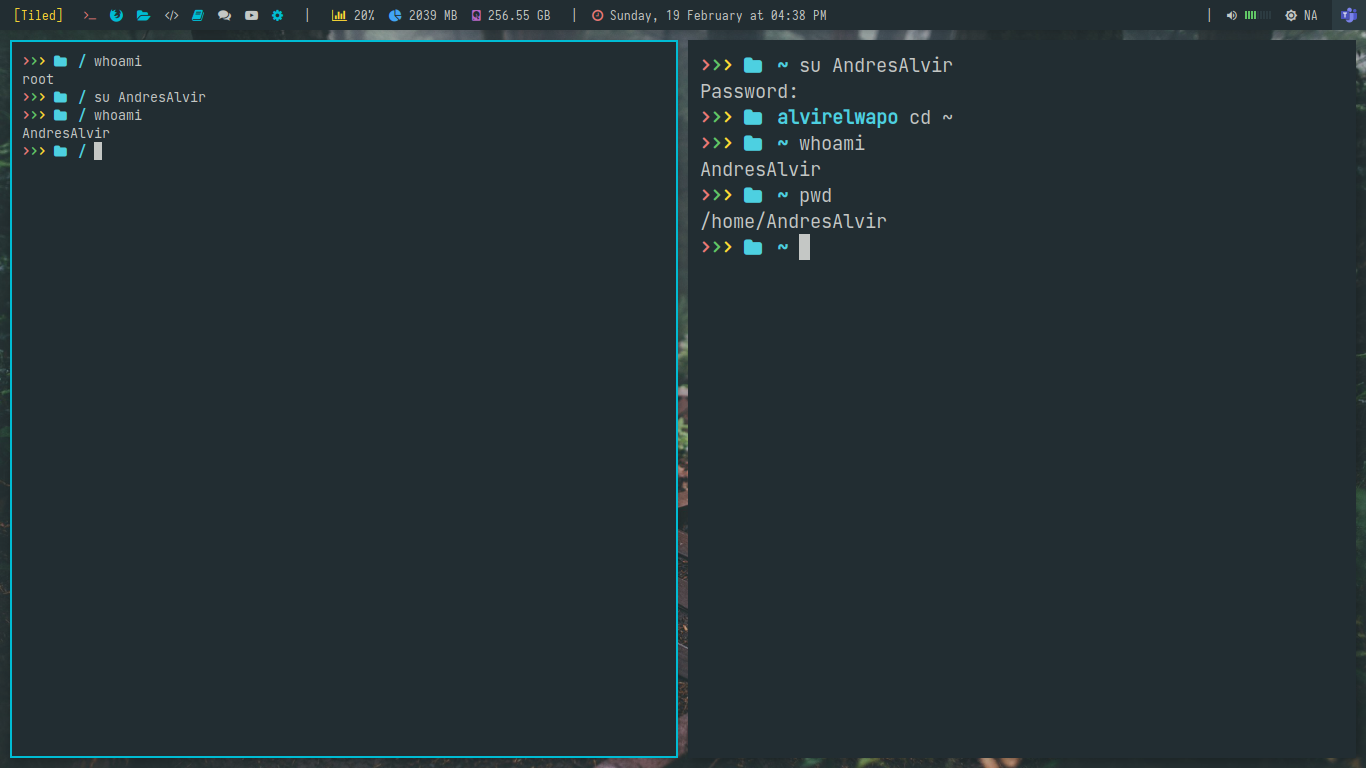
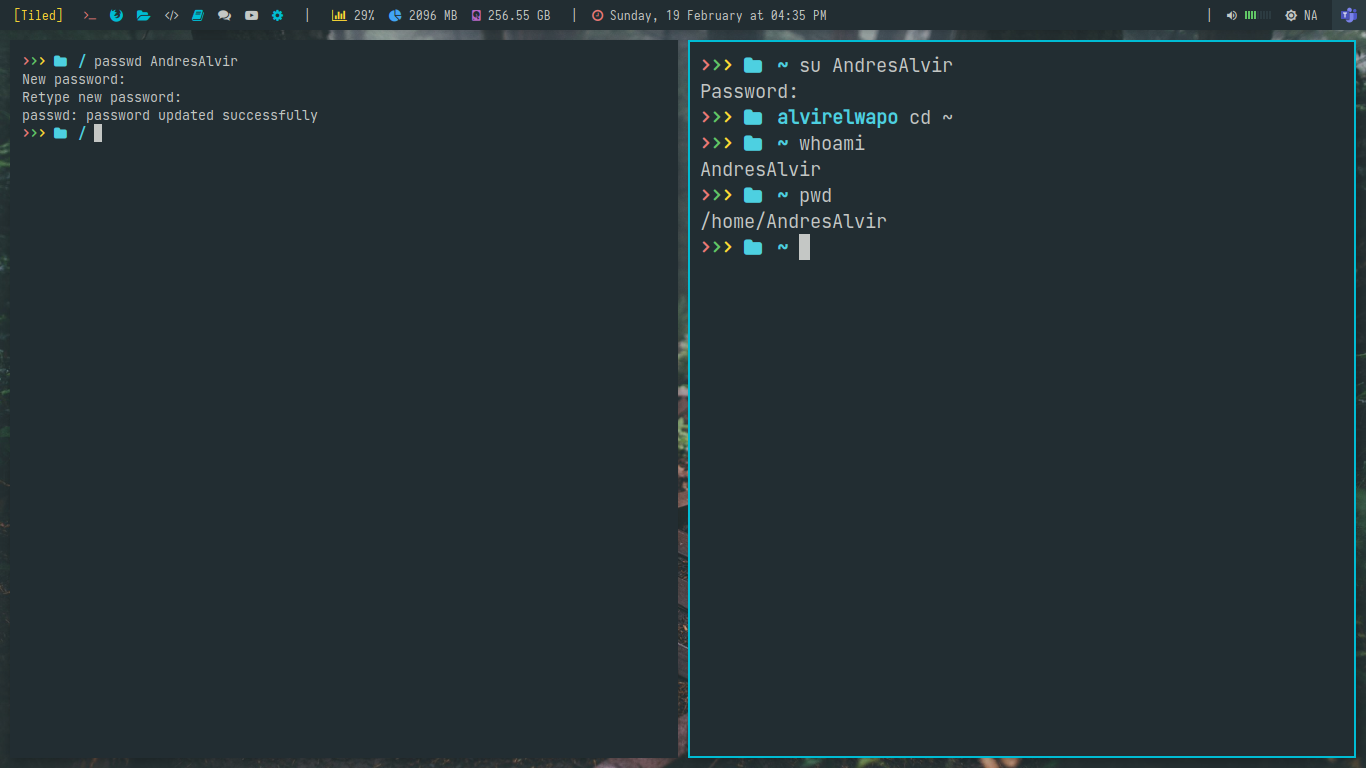
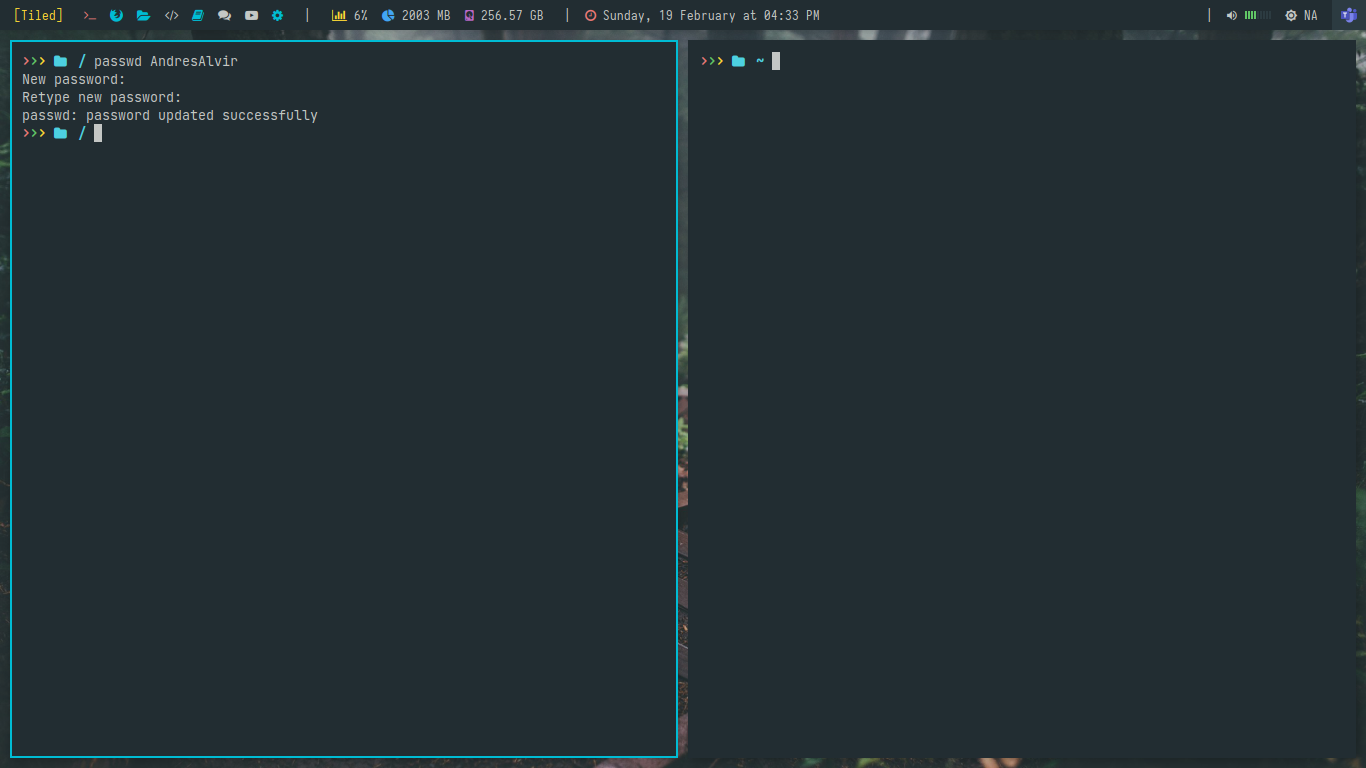
Luego me dí cuenta que un gran ejemplo de un sistema capaz de Autenticar, Autorizar y Registrar todos los movimientos de uno o varios usuarios era cualquier sistema GNU-Linux, el cual usaré para ambos ejemplos.

Ejemplo personal.

Para el primer paso de la práctica personal debo, en primera instancia, tener una cuenta de usuario en Linux que sea la “administradora” o “root”, y otra normal que será administrada por “root”, en este caso se llamará Andrés Alvir.



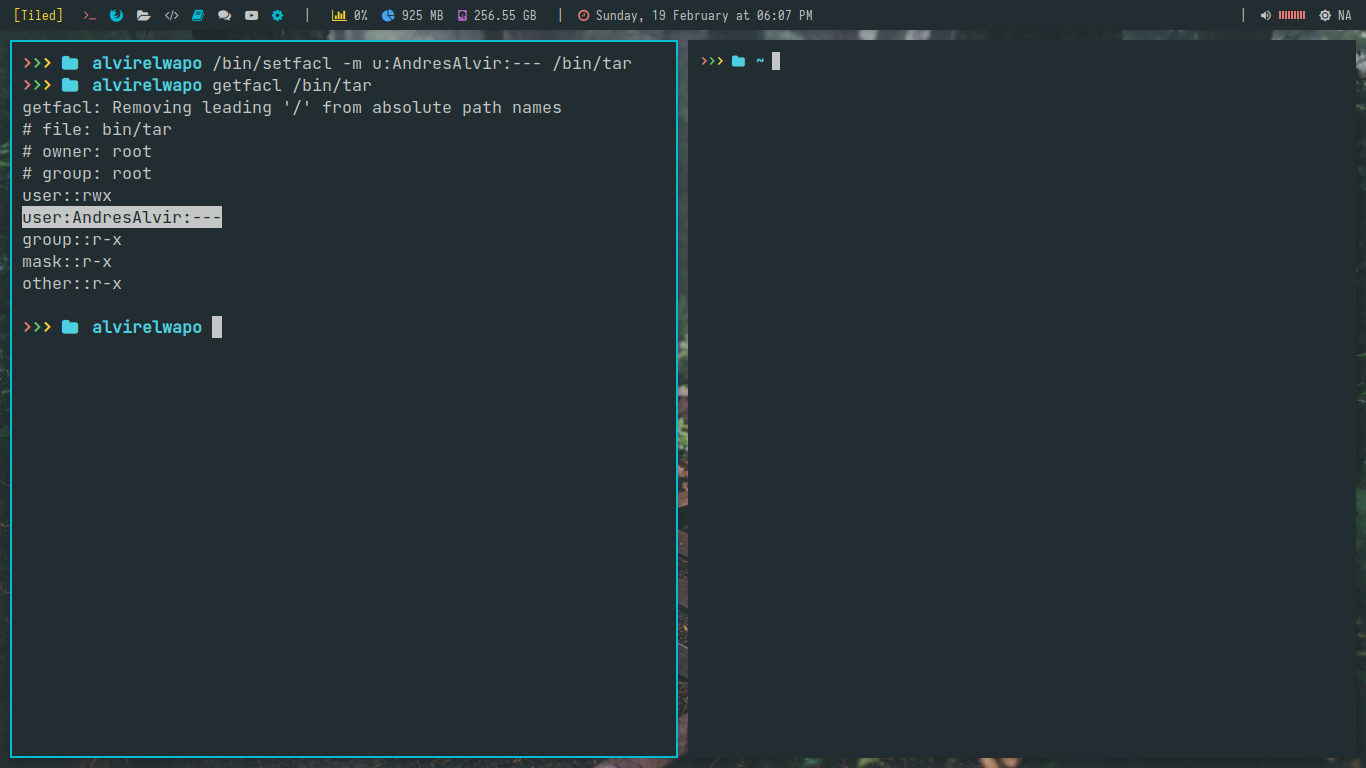
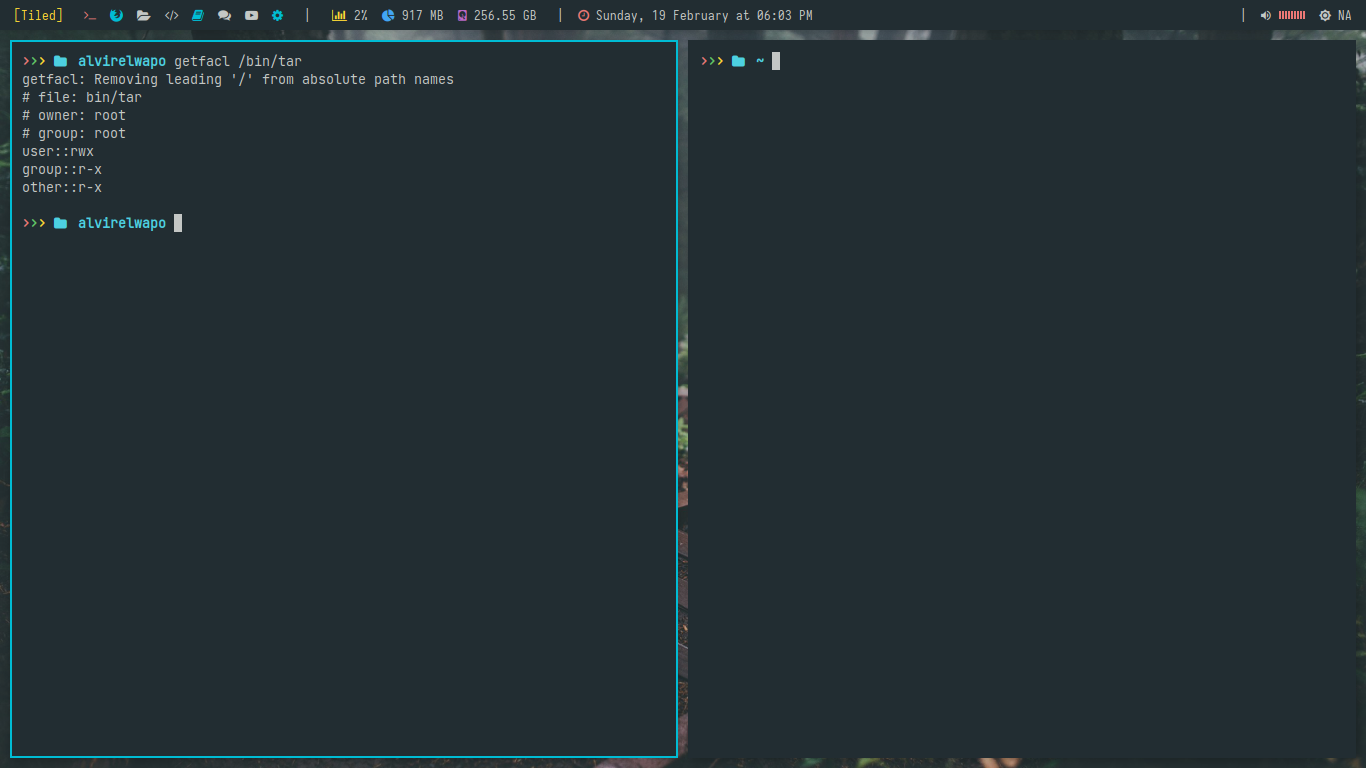
Para la implementación de “Autenticación”, debemos asignarle a una contraseña a AndresAlvir, para que pueda ingresar a su cuenta de usuario.



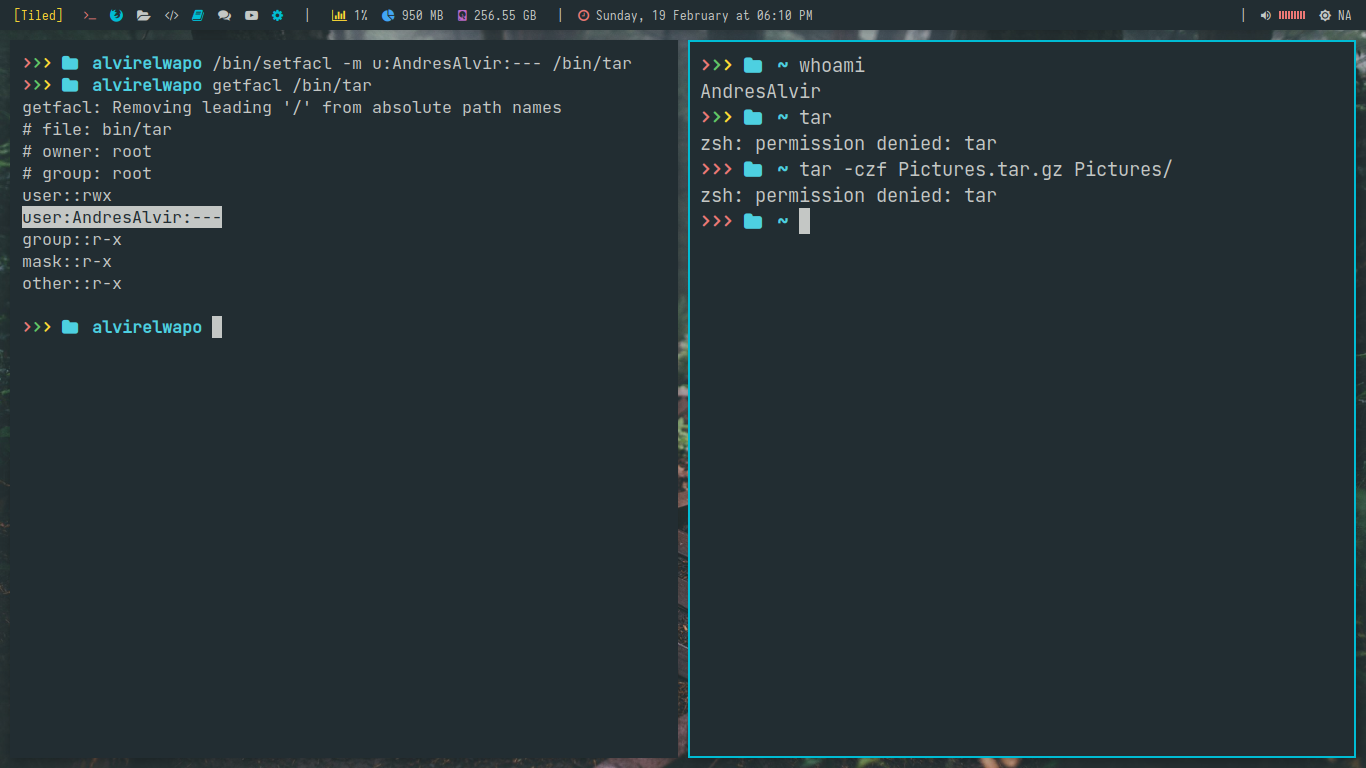
Con esto tenemos cubierta la primer parte de nuestro ejemplo, Autenticación. Hay 2 formas de ingresar a nuestro usuario AndresAlvir, la primera es desde una terminal, utilizando el comando Switch User (su) seguido del nombre de usuario al que queremos ingresar (su AndresAlvir) o desde la cuenta root, sin necesidad de una contraseña ya que root administra todas las cuentas.

Como segundo punto tenemos Authorization, siendo administradores de nuestra máquina Linux, utilizando y modificando las ACL (Access Control Lists), podemos cambiar algunas cosas a las que tiene acceso AndresAlvir:

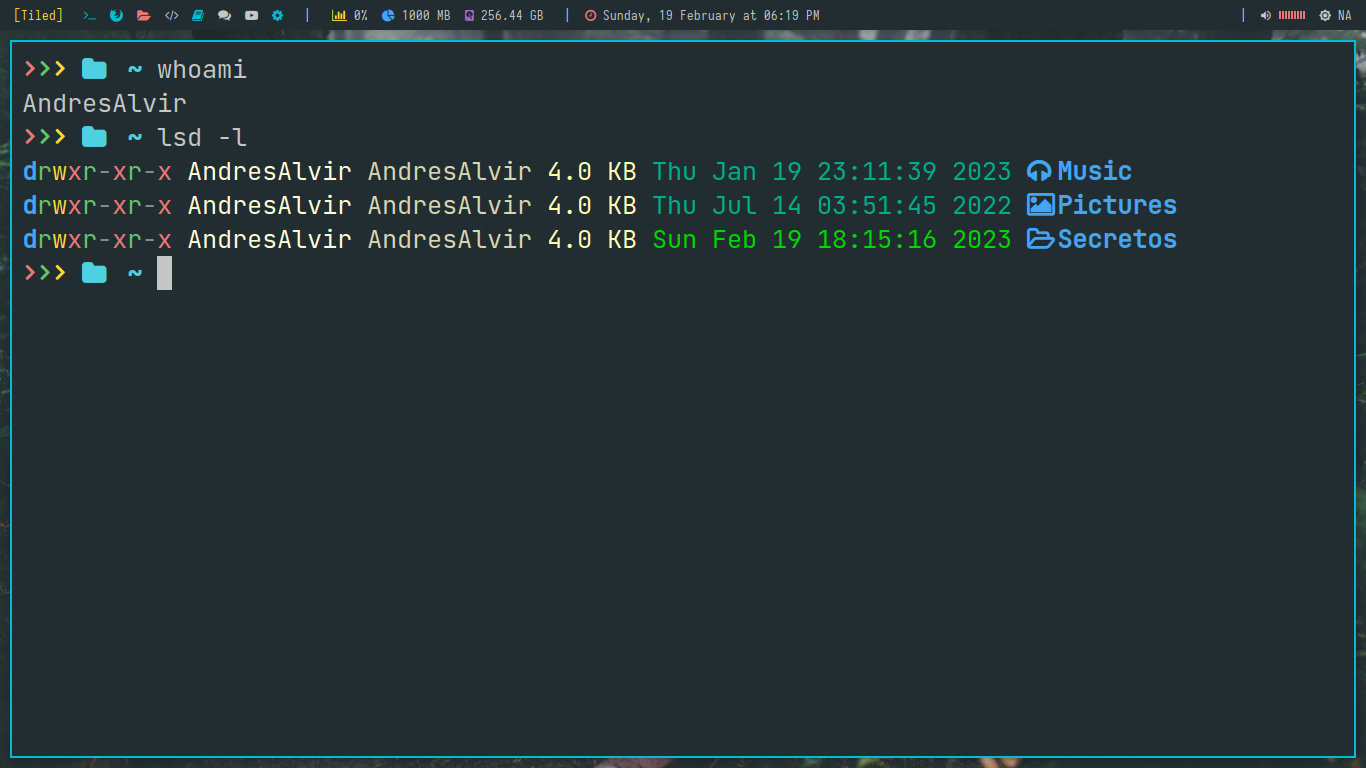
Utilizando los comandos getfacl y setfacl, podemos ver y modificar las listas de acceso a cada comando que tengamos en mente, para el ejemplo utilizaremos “tar”.



Como podemos observar, para el usuario AndresAlvir, los permisos (read, write, execute) son - - -. Lo cual quiere decir que no hay manera que desde el usuario Andres Alvir se acceda a ese comando.



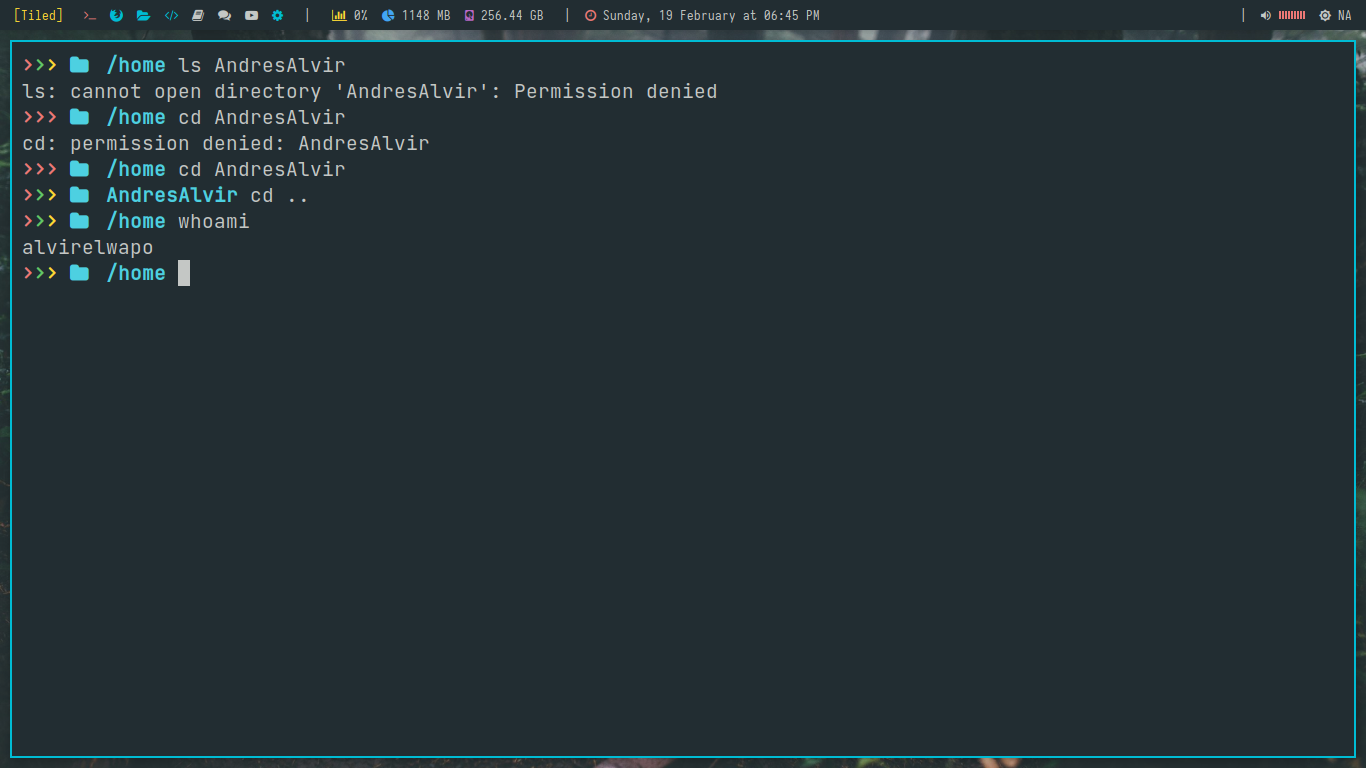
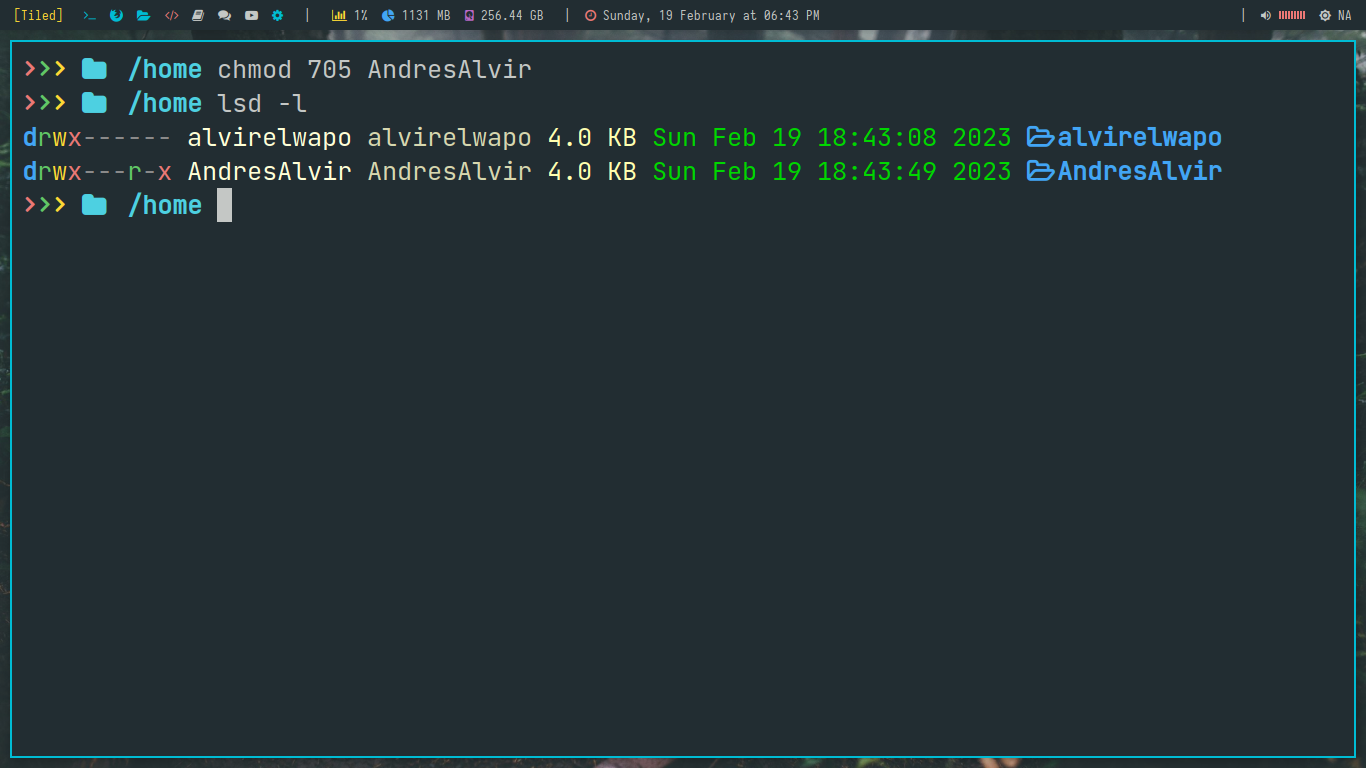
De igual manera, el usuario AndresAlvir puede proteger sus archivos de otros usuarios en el sistema a través de los permisos.



Como podemos observar, el usuario Andrés Alvir tiene en su carpeta principal (/home/AndresAlvir), 3 carpetas, Music, Pictures y Secretos.

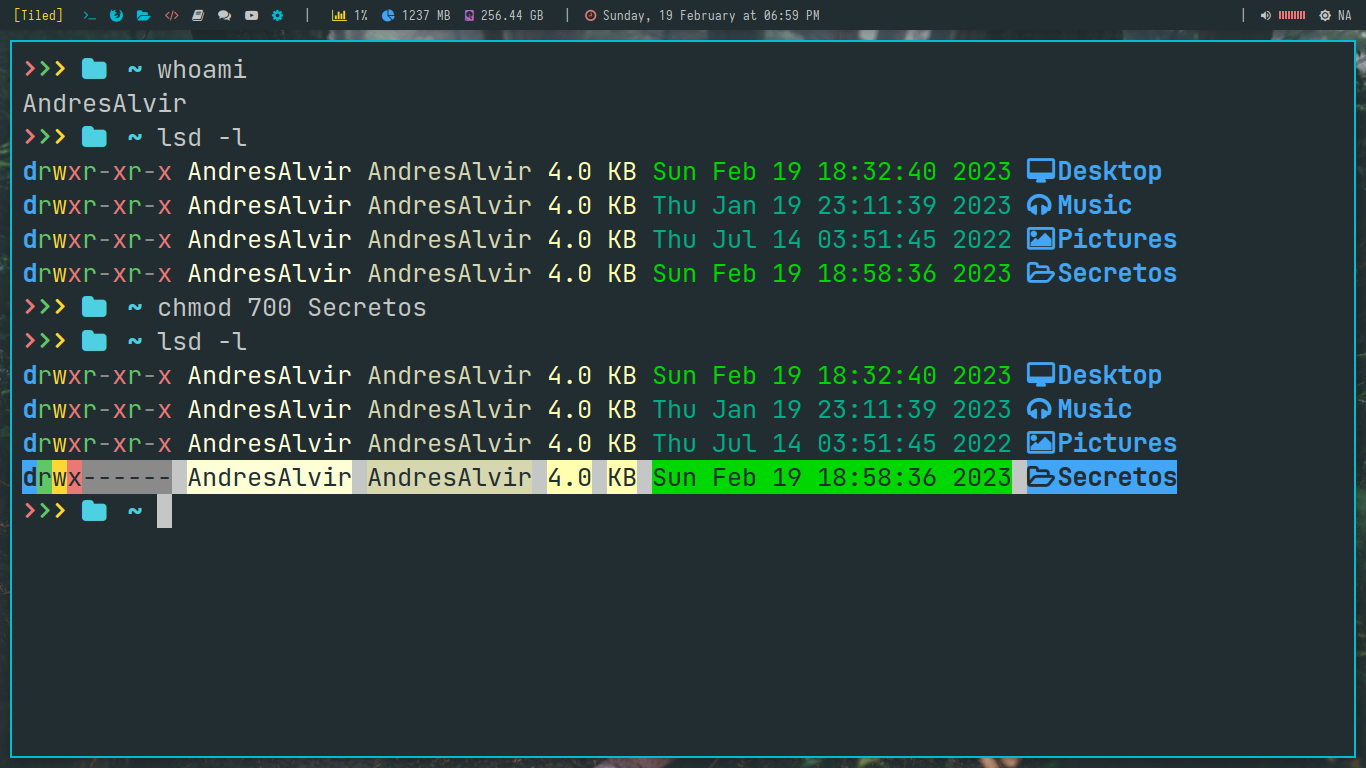
Para el ejemplo ocuparé un usuario secundario llamado alvirelwapo, para este caso plantearemos que AndresAlvir y alvirelwapo colaboran en un proyecto llamado IGGY, del cual los archivos se encuentran dentro de las carpetas del ususario AndresAlvir (~/Desktop/IGGY).

Primero, AndresAlvir da permisos para que otros puedan acceder a su carpeta con cd.

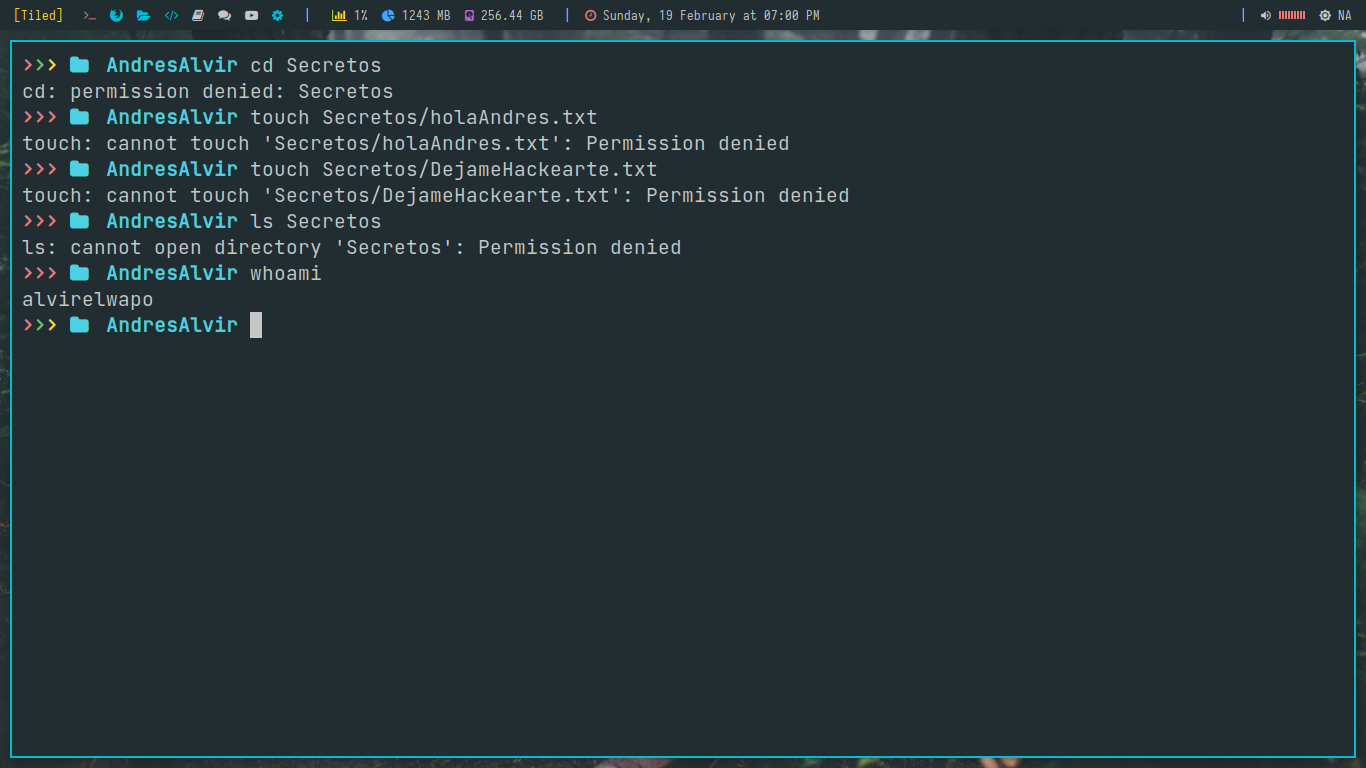


Aquí se ve el cambio que se realizó arriba

El segundo paso, es proteger su carpeta de Secretos, AndresAlvir no quiere que conozcan sus secretos. Para esto, utilizaremos el comando chmod, haciendo que solo el usuario dueño de la carpeta pueda visualizar los contenidos.

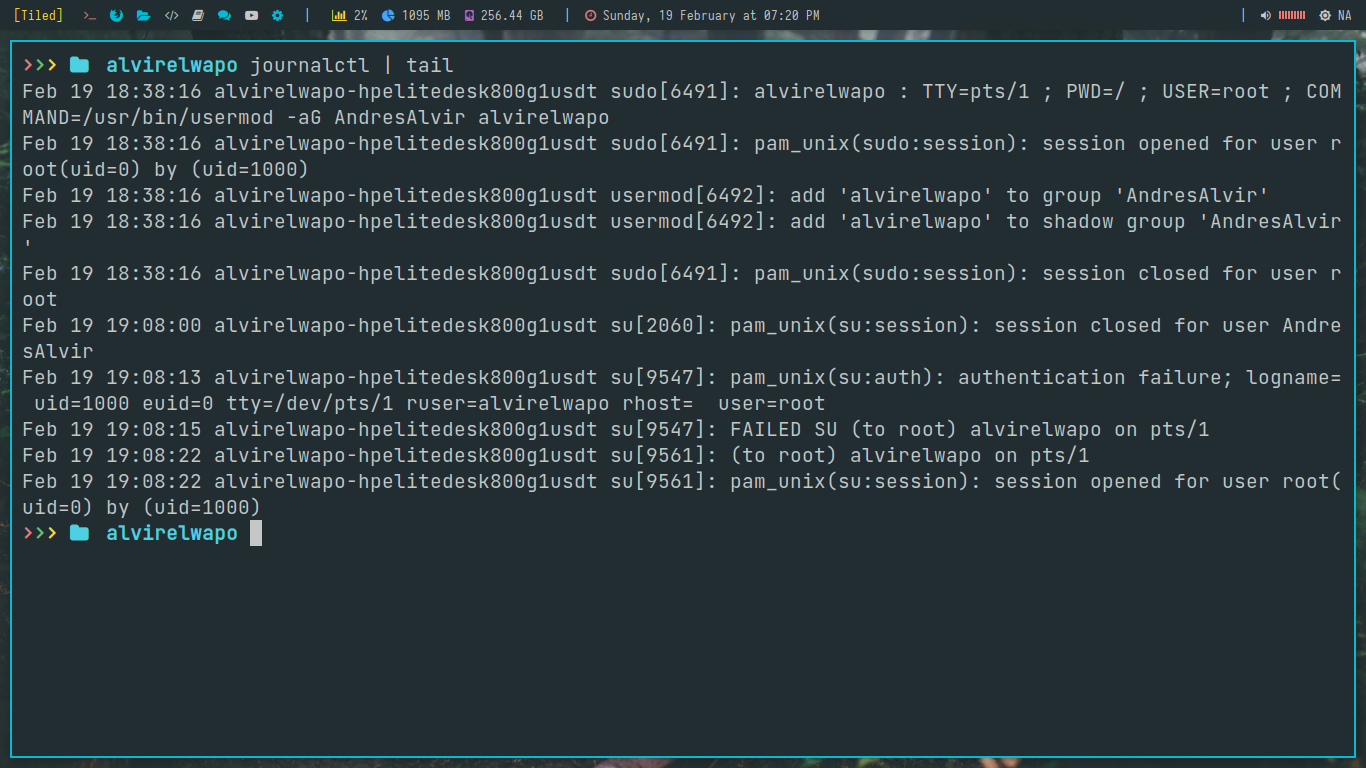
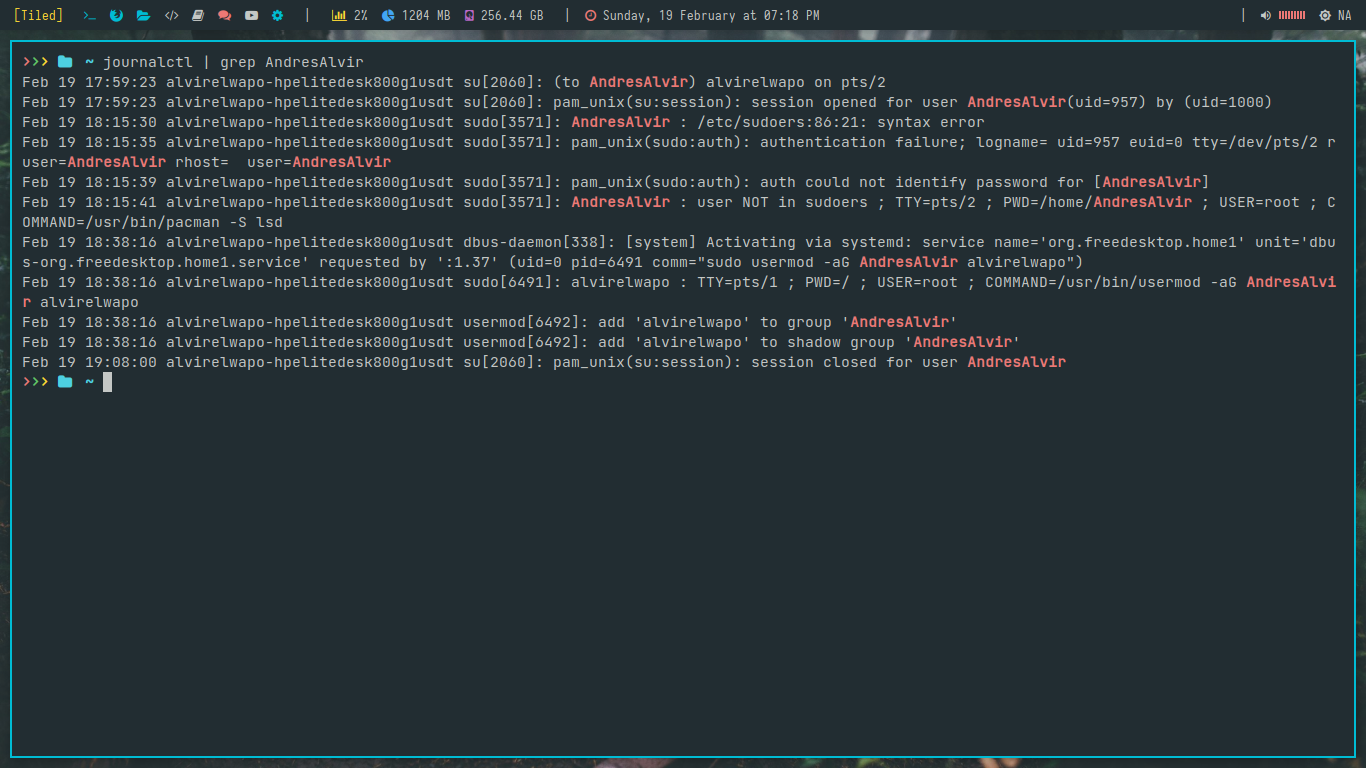


Y listo, Ahora no podemos hacer nada con los archivos de Secretos desde el usuario alvirelwapo…



Estas son algunas de las formas en las que podemos controlar las cosas que se realizan en el sistema.

Para la tercera porción de nuestro ejemplo, tenemos que contarizar todo lo que realizamos. Afortunadamente, Linux almacena casi todo lo que se realiza dentro del sistema…



Ejemplo ChatGPT

Primer paso: preguntarle a ChatGPT qué es AAA:

<YO>: can you explain Authentication Authorization and Accounting

<ChatGPT>: Certainly!

Authentication, Authorization, and Accounting, often referred to as AAA, are three important concepts in computer security that help protect and control access to resources.

Here's a brief explanation of each:

1. Authentication: Authentication is the process of verifying the identity of a user or entity that is trying to access a system or resource. It's important to ensure that only authorized users can gain access to the system, and authentication is the first step in achieving this. There are various methods of authentication, including passwords, biometrics, smart cards, and more.
2. Authorization: Once a user's identity has been authenticated, the next step is to determine what actions they are authorized to perform. Authorization is the process of determining whether a user has the necessary permissions to access a particular resource or perform a specific action. Authorization is typically based on a set of rules or policies that determine what actions a user can or cannot perform.
3. Accounting: Accounting involves keeping track of all user activity on a system or resource. This includes logging user actions, recording resource usage, and generating reports that can be used for auditing and compliance purposes. Accounting provides an important record of who did what on a system, which can be useful for troubleshooting, forensics, and monitoring compliance.

In summary, authentication is the process of verifying a user's identity, authorization is the process of determining what actions they are authorized to perform, and accounting is the process of tracking user activity on the system. Together, these three concepts are critical for securing computer systems and ensuring that only authorized users can access and use valuable resources.

Luego, le pregunté si podría hacerme un ejemplo utilizando comandos de Bash…

<YO>: Can you do a practical example using bash?

Sure, I can provide a practical example using Bash. Let's use a simple scenario to demonstrate the concepts of authentication, authorization, and accounting.

<ChatGPT>: Suppose you have a Linux server that you want to secure. You have a user account named "johndoe" that you want to use to manage the server. You want to make sure that only authorized users can access the server, that "johndoe" has the necessary permissions to manage the server, and that you can track all activity on the server for auditing purposes.

Here's how you might implement authentication, authorization, and accounting in Bash:

1. Authentication: To authenticate "johndoe" and ensure that only authorized users can access the server, you can use the "passwd" command to set a password for the "johndoe" account. This will require anyone who wants to log in to the server to provide the correct username and password combination. Here's how you might set a password for "johndoe" using the "passwd" command:

#######> sudo passwd johndoe

1. Authorization: To ensure that "johndoe" has the necessary permissions to manage the server, you can add them to the "sudoers" file, which allows them to run privileged commands using the "sudo" command.

Here's how you might add "johndoe" to the "sudoers" file:

#####> sudo visudo

This will open the "sudoers" file in a text editor. You can then add the following line to give "johndoe" full sudo access:

#####> johndoe ALL=(ALL) ALL

1. Accounting:

To track all activity on the server, you can enable logging using the "syslog" daemon. This will record all user activity, including logins, commands executed, and other system events.

Here's how you might enable logging using "syslog":

#####> sudo nano /etc/rsyslog.conf

This will open the syslog configuration file in a text editor. You can then add the following line to enable logging:

#####> auth,authpriv.\* /var/log/auth.log

Save and close the file. Restart the "rsyslog" service to apply the changes:

#####> sudo service rsyslog restart

Now, all user activity on the server will be logged to the "/var/log/auth.log" file, which you can use for auditing and compliance purposes.

In summary, by setting a password for the "johndoe" account, adding them to the "sudoers" file, and enabling logging using "syslog", you've implemented authentication, authorization, and accounting on your Linux server.

<YO>: you're the best chatgpt

<ChatGPT>: Thank you for the kind words! I'm always here to help if you have any more questions.

|  |
| --- |
| **Fuentes Consultadas** |

https://www.thegeekdiary.com/how-to-disable-a-specific-command-for-a-specific-user-in-linux/

https://www.computerhope.com/jargon/a/acl.htm

https://www.guru99.com/file-permissions.html