Ámbito de aplicación

Este procedimiento es aplicable por cualquier usuario que necesite usar el método https implementado por git.

Propósito del documento

Este documento describe los pasos a seguir de como autentificar en git mediante https.

Git usa dos métodos para ello: ssh y https. El más usual es ssh ya que usa criptografía asimétrica para autenticación (sin contraseña). Sin embargo, si la salida del puerto TCP 22 no está permitida, no se puede usar este método y se tiene que optar por HTTPS, que usa el puerto TCP 443, que está permitido habitualmente.

Para hacer uso de este segundo método, es necesario proporcionar manualmente el usuario y la contraseña de git (gitlab/github...) cada vez, lo cuál puede ser un problema a la hora de usar de forma contínua git en línea de comandos.

Para solucionar esto, git dispone de "credentials.helper", que permite guardar en caché los datos de autenticación. El tiempo durante el que están vigentes estos datos se puede configurar en git, pero si lo que se quiere es que nunca más pida usuario y contraseña manualmente, se puede usar el argumento "store", quedando el comando a ejecutar así:

git config credentials.helper store

lo que creará un archivo llamado .git-credentials o bien .git-credentials-NOMBRE\_REPOSITORIO.

Este archivo está en texto plano, así que cualquiera que tenga acceso a dicho directorio podrá verlo. Hay que tenerlo en cuenta a la hora de usar este método.

Instrucciones técnicas

**Clonar el Repositorio**

* Descripción breve del proceso de clonado del repositorio utilizando la URL HTTPS proporcionada.
* Información sobre la introducción de credenciales.

**Acceder al Repositorio**

* Detalles sobre cómo acceder a la carpeta del repositorio clonado.
* Instrucciones para navegar a la ruta del repositorio.

**Configurar Git para Almacenar Credenciales**

* Explicación de cómo configurar Git para almacenar credenciales dentro del repositorio.
* Introducción del comando específico utilizado para esta configuración.
* Detalle sobre las opciones y parámetros del comando.

**Realizar un Push para Crear el Archivo de Credenciales**

* Proceso para realizar un push o una operación similar que requiera autenticación.
* Creación del archivo de credenciales al introducir las credenciales solicitadas.
* Confirmación de que no será necesario introducir las credenciales nuevamente para este repositorio.

Pasos a seguir:

## Paso 1: Clonar el repositorio.

**-Clonamos nuestro repositorio e introducimos las credenciales.**

Git clone [url https de nuestro repositorio]

**Ejemplo:**

Git clone https://gitlab.com/usuairo-de-git/ejemplo.git

## Paso 2: Acceder al repositorio.

**-Accedemos dentro de la carpeta de nuestro repositorio.**

cd [ruta de nuestro repositorio]

**Ejemplo**

cd /git/ejemplo

## Paso 3: Configurar Git para Almacenar Credenciales.

**-Nos ubicamos dentro de nuestro repositorio e introduciremos este comando.**

git config credential.helper 'store --file .git-credentials-nombrerepo'

**Explicación del comando.**

**1.** **git config:** Este es el comando principal de Git para configurar opciones. Se utiliza para establecer variables de configuración que controlan el comportamiento de Git.

**2.** **credential.helper:** Este es el nombre de la configuración que estamos ajustando. Los ayudantes de credenciales (credential helpers) son programas que Git usa para almacenar y recuperar credenciales de usuario. Hay varios ayudantes disponibles, como cache, store, y osxkeychain, entre otros.

**3. 'store --file ~/.git-credentials-nombrerepo':** Esta es la opción específica que estamos pasando al helper de credenciales. Desglosemos esto:

**> store:** Indica que vamos a usar el helper de credenciales de tipo "store". Este helper guarda las credenciales en un archivo de texto sin cifrar.

> --file ~/.git-credentials-nombrerepo: Esta opción específica le dice al helper de tipo "store" que use un archivo en particular para guardar las credenciales. En este caso, el archivo se ubicará en el directorio home del usuario y se llamará .git-credentials-nombrerepo**.**

**\* Se podría también usar el mismo comando pero con ruta absoluta pero las credenciales se guardaran en nuestro home**

git config credential.helper 'store --file ~/.git-credentials-nombrerepo'

## Paso 4: Realizar un Push para crear el archivo de credenciales.

-Con el comando ya introducido, deberemos hacer un push o similar que pida de nuevo las credenciales, cuando las introduzcamos se nos creara el archivo .git-credentials-nombrerepo, ya generado este archivo ya no tendremos que introducir las credenciales en ese repositorio.

Git push

## Conclusión

Siguiendo estos pasos, habrás configurado Git para almacenar tus credenciales de manera que no tengas que introducirlas repetidamente cada vez que interactúes con el repositorio.

|  |  |
| --- | --- |
| **Creado por** | *Israel Vara Box* |
| **Fecha Creación** | *2024-05-29* |
| **Revisado por** |  |
| **Fecha Última Revisión** |  |