Monitorizar Ubuntu Server

Bot de Telegram

Miguel Ángel Roldán de Haro

10 de enero del 2025

Índice

Descripción del proyecto

Este proyecto implementa un **bot de Telegram** diseñado para monitorear, gestionar y reforzar la seguridad de un **servidor Ubuntu** que ejecuta Proxmox o cualquier entorno similar. El bot permite al administrador del sistema interactuar directamente con el servidor desde Telegram, proporcionando herramientas avanzadas de supervisión y control en tiempo real.

El bot se enfoca en:

- **Monitoreo del Servidor**:
    - Proporciona información en tiempo real sobre el uso de recursos (CPU, RAM, disco).
    - Muestra un listado de usuarios conectados con detalles relevantes (nombre, terminal, host, tiempo de conexión).
    - Lista los procesos que más consumen recursos.
- **Gestión de Conexiones**:
    - Permite bloquear o desbloquear usuarios específicos.
    - Implementa un **modo seguro** que bloquea todas las conexiones entrantes, útil en caso de amenazas o mantenimiento.
- **Ciberseguridad**:
    - Detecta usuarios bloqueados intentando acceder y les muestra un mensaje personalizado indicando que su acceso está restringido.
    - Informa a los usuarios sobre el estado del servidor (bloqueado/operativo) con mensajes dinámicos.
    - Integra controles de seguridad directamente en el flujo de conexiones SSH mediante el uso de PAM (Pluggable Authentication Modules).
- **Automatización y Conveniencia**:
    - Integra comandos simples de Telegram para controlar el servidor desde cualquier lugar
    -

Requisitos Previos

Hardware

- Un **servidor Ubuntu** con acceso a la terminal.
- **Telegram** instalado para interactuar con el bot.

Software

1. **Actualizar el sistema operativo:**

   sudo apt update && sudo apt upgrade -y

   ```
   root@ubuntu-server:~# sudo apt update && sudo apt upgrade -y
   ```

2. **Instalar Python y pip:**

   sudo apt install python3 python3-pip -y

   ```
   root@ubuntu-server:~# sudo apt install python3 python3-pip -y
   ```

3. **Instalar dependencias necesarias para el proyecto:**

   pip3 install python-telegram-bot psutil

   ```
   root@ubuntu-server:~# pip3 install python-telegram-bot psutil
   ```

   pip3 install pyTelegramBotAPI

   ```
   root@ubuntu-server:/bin# pip3 install pyTelegramBotAPI
   ```

4. **Instalar IP tables:**

   sudo apt install iptables

   ```
   root@ubuntu-server:~# sudo apt install iptables
   ```

5. **Verificar el acceso SSH configurado:**

   Asegúrate de que el archivo /etc/ssh/sshd_config esté configurado correctamente para permitir conexiones SSH.

```
root@ubuntu-server:~# cat /etc/ssh/sshd_config
```

Configuración del Entorno

Editar el archivo sudoers y añadir esto:

tu_usuario ALL=(ALL) NOPASSWD: /sbin/iptables, /bin/systemctl, /usr/bin/who, /usr/bin/free, /bin/df

```
root ALL=(ALL) NOPASSWD: /sbin/iptables, /bin/systemctl, /usr/bin/who, /usr/bin/free, /bin/df
```

Crear el archivo de usuarios bloqueados:

sudo touch /etc/security/bloqueados.txt

```
root@ubuntu-server:~# sudo touch /etc/security/bloqueados.txt
```

sudo chmod 600 /etc/security/bloqueados.txt

```
root@ubuntu-server:~# sudo chmod 600 /etc/security/bloqueados.txt
```

Crear el archivo de modo seguro:

sudo touch /etc/security/seguro.txt

```
root@ubuntu-server:/# sudo touch /etc/security/seguro.txt
```

sudo chmod 600 /etc/security/seguro.txt

```
root@ubuntu-server:/# sudo chmod 600 /etc/security/seguro.txt
```

Creamos el script para manejar accesos: Crea el archivo custom_ssh_check.sh en /usr/local/bin/:

sudo nano /usr/local/bin/custom_ssh_check.sh

```
root@ubuntu-server:/usr/local/bin# sudo nano /usr/local/bin/custom_ssh_check.sh
```

Pegamos el contenido del script, luego otorgamos permisos de ejecución:

sudo chmod +x /usr/local/bin/custom_ssh_check.sh

```
root@ubuntu-server:/usr/local/bin# sudo chmod +x /usr/local/bin/custom_ssh_check.sh
```

Edita el archivo de configuración de PAM para SSH:

sudo nano /etc/pam.d/sshd

```
root@ubuntu-server:/usr/local/bin# sudo nano /etc/pam.d/sshd
```

Añade la siguiente línea al inicio del archivo:

auth required pam_exec.so seteuid /usr/local/bin/custom_ssh_check.sh

Configurar PAM para el control de accesos SSH: Edita el archivo /etc/pam.d/sshd y añade la línea al inicio:

auth required pam_exec.so seteuid /usr/local/bin/custom_ssh_check.sh

```
  GNU nano 6.2                                /etc/pam.d/sshd
# PAM configuration for the Secure Shell service
auth required pam_exec.so seteuid /usr/local/bin/custom_ssh_check.sh
```

Reinicia el servicio SSH para aplicar cambios:

sudo systemctl restart sshd

```
root@ubuntu-server:/usr/local/bin# sudo systemctl restart sshd
```

Configuración del Bot de Telegram

1. **Crear el bot en BotFather y obtener el token:**
    - Ejecuta el comando /newbot en BotFather en Telegram.
    - Copia el token proporcionado.
2. **Configurar el token en el script del bot: En el archivo bot.py, reemplaza "TU_TOKEN_AQUÍ" por el token obtenido:**
    - BOT_TOKEN = "1234567890EJEMPLO"

```
# TOKEN del bot de Telegram (proporcionado por BotFather)
TOKEN = 'TU_TOKEN_DE_TELEGRAM'
bot = telebot.TeleBot(TOKEN)
```

3. **Insertar el ID de usuario de telegram (se puede consultar gracias al bot @userinfobot de telegram)**

```
# Variables de estado
modo_seguro = False
bloqueados = []  # Lista de IPs bloqueadas
admin_id = TU_CHAT_ID  # Reemplaza con tu ID de Telegram para permisos de administración
```

Funcionalidades del Bot

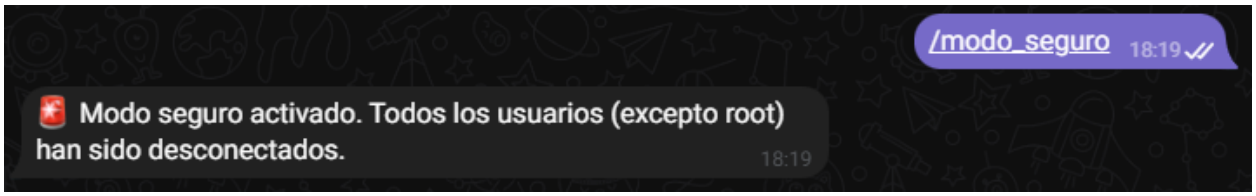
Comandos Disponibles:

- **/start -** Muestra este mensaje de bienvenida
- **/help -** Muestra esta lista de comandos
- **/modo_seguro -** Activa el modo seguro
- **/desactivar_seguro -** Desactiva el modo seguro
- **/listar_bloqueados -** Muestra usuarios bloqueados
- **/listar_no_bloqueados -** Muestra usuarios no bloqueados
- **/top_procesos -** Muestra el top 10 de procesos
- **/recursos -** Muestra información de recursos
- **/usuarios -** Lista usuarios conectados

Gestión de Accesos y Seguridad

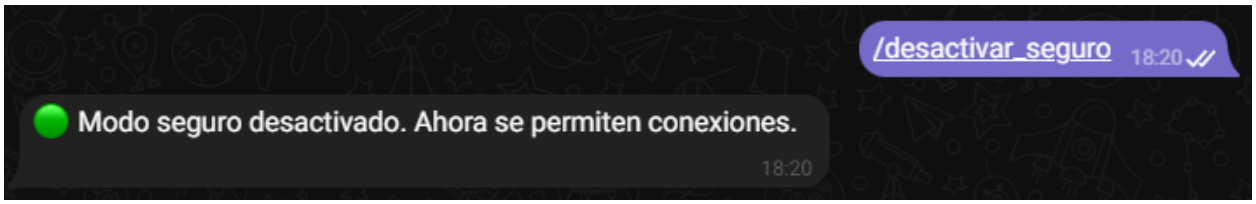
Prueba de /modo_seguro: Activación del modo seguro (se deshabilitan las conexiones ssh):

/modo_seguro



Prueba de /desactivar_seguro: Desactivación del modo seguro (se habilitan las conexiones ssh):

/desactivar_seguro

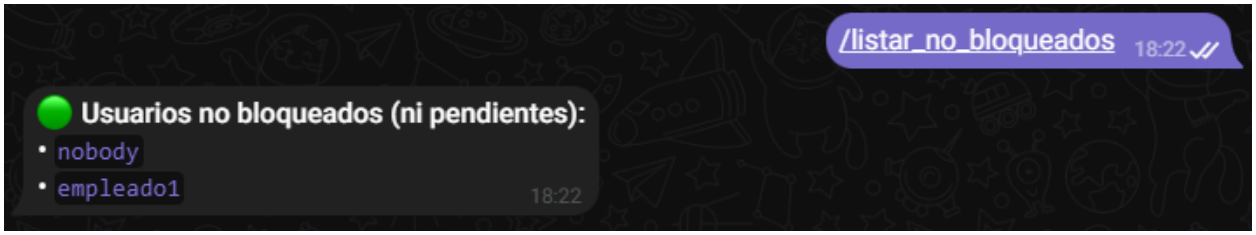


Prueba de /listar_bloqueados: Listar todos los usuarios bloqueados:

/listar_bloqueados

Prueba de/listar_no_bloqueados: Listar todos los usuarios no bloqueados:

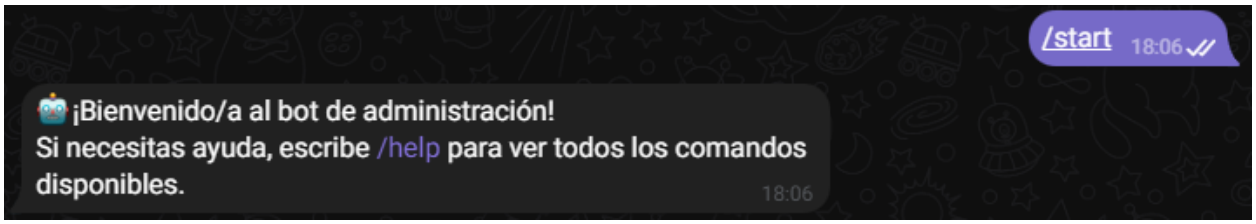
/listar_no_bloqueados



Monitoreo del Servidor

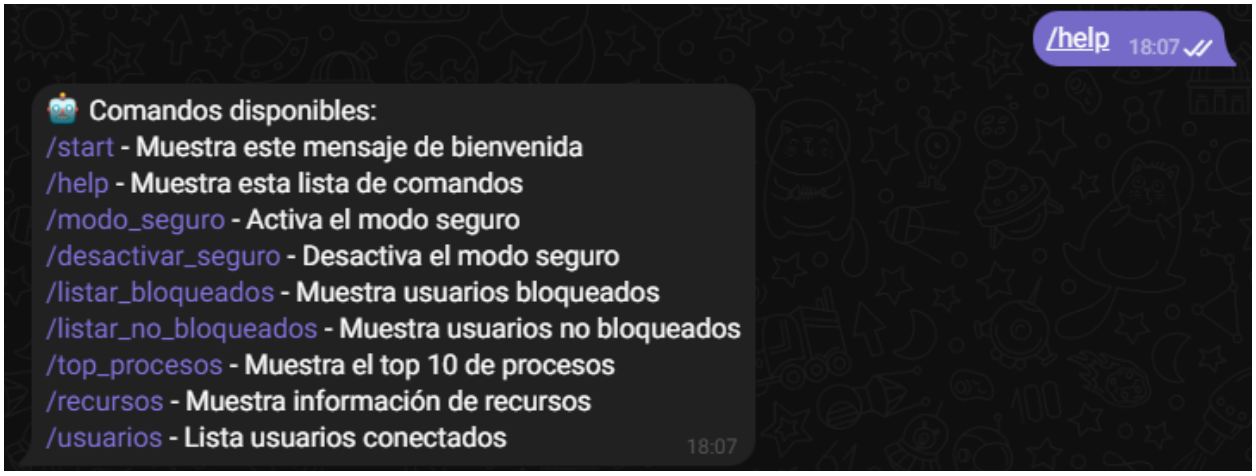
Prueba de /start: Comenzar la ejecución:

/start



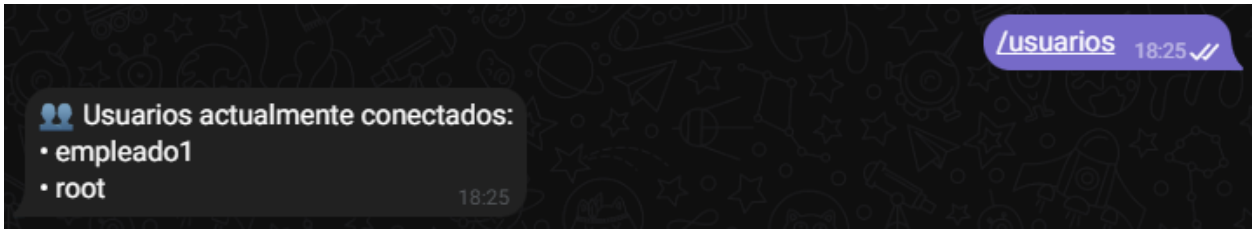
Prueba de /help: Ver todos los comandos disponibles:

/help



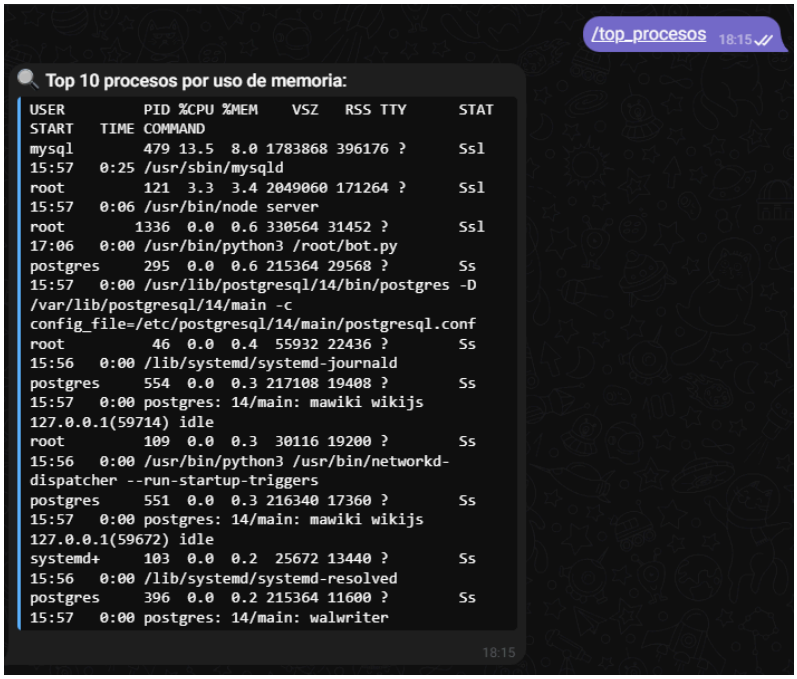
Prueba de /usuarios: Conéctate al servidor con diferentes usuarios y usa:

/usuarios



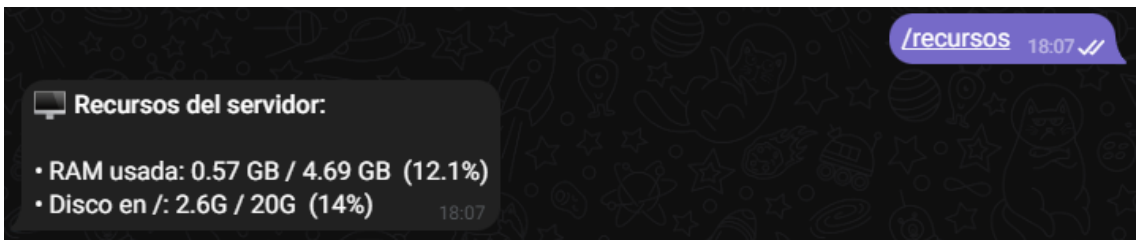
Prueba de /top_procesos: Valida que el bot reporte los procesos que más consumen recursos:

/top_procesos



Prueba de /recursos: Asegúrate de que el bot muestre el uso de RAM y disco:

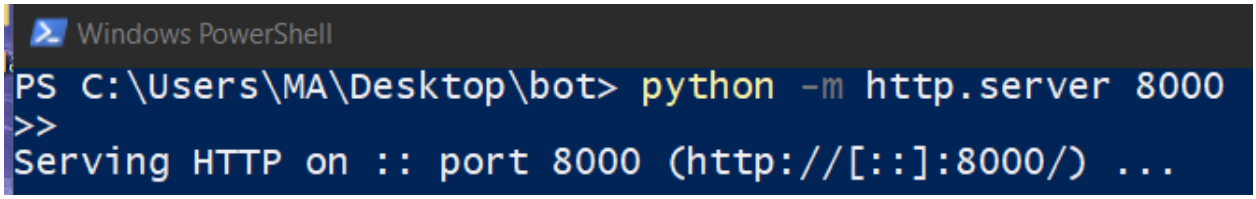
/recursos



Configuración Automática

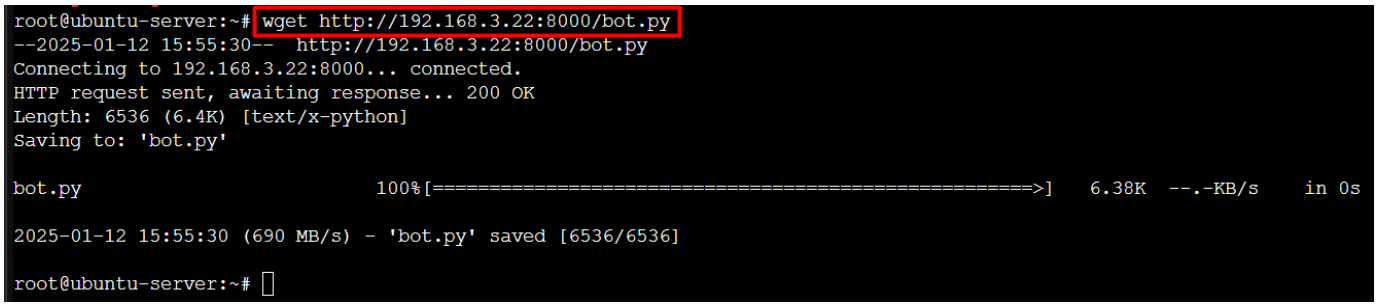
Insertar el script de python 3 del bot en el servidor

Desde mi equipo windows voy a transferir el archivo al servidor por medio de uso de un servidor http python3

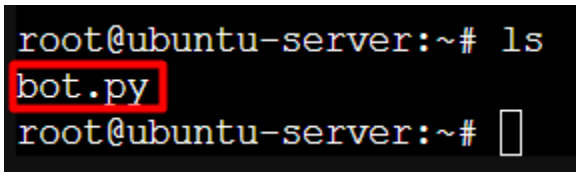


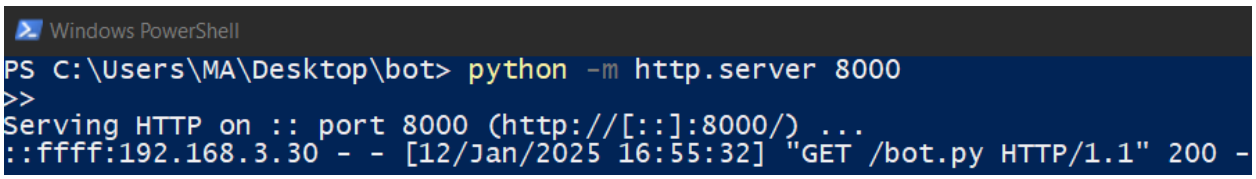
Ahora desde el servidor descargamos el recurso

wget http://192.168.3.22:8000/bot.py



Vemos que el recurso se ha descargado con éxito





Dar permisos de ejecución:

sudo chmod +x bot.py

```
root@ubuntu-server:~# sudo chmod +x bot.py
```

Crear un servicio de systemd para ejecutar el bot:

Archivo del servicio:

sudo nano /etc/systemd/system/telegram-bot.service

```
root@ubuntu-server:~# sudo nano /etc/systemd/system/telegram-bot.service
```

Contenido:

[Unit]

Description=Bot de Telegram para monitoreo y seguridad del servidor

After=network.target

[Service]

ExecStart= /root/bot.py

Restart=always

User=root

Group=root

[Install]

WantedBy=multi-user.target

```
  GNU nano 6.2                                /etc/systemd/system/telegram-bot.service
[Unit]
Description=Bot de Telegram para monitoreo y seguridad del servidor
After=network.target

[Service]
ExecStart=/usr/bin/python3 /root/bot.py
Restart=always
User=root
Group=root

[Install]
WantedBy=multi-user.target
```

Habilitar y ejecutar el servicio:

sudo systemctl daemon-reload

```
root@ubuntu-server:~# sudo systemctl daemon-reload
```

sudo systemctl enable telegram-bot

```
root@ubuntu-server:~# sudo systemctl enable telegram-bot
Created symlink /etc/systemd/system/multi-user.target.wants/telegram-bot.service -> /etc/systemd/system/telegram-bot.service.
```

sudo systemctl start telegram-bot

```
root@ubuntu-server:~# sudo systemctl start telegram-bot
```

sudo systemctl status telegram-bot

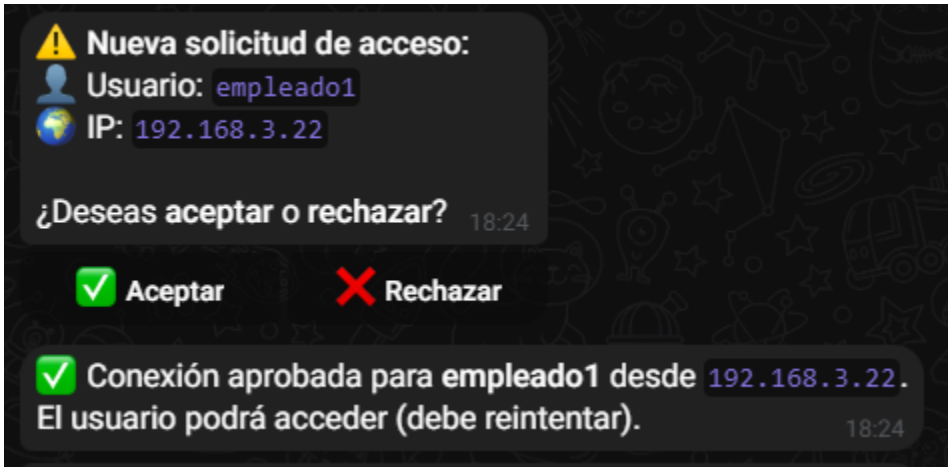
```
root@ubuntu-server:~# sudo systemctl status telegram-bot
* telegram-bot.service - Bot de Telegram para monitoreo y seguridad del servidor
     Loaded: loaded (/etc/systemd/system/telegram-bot.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-01-12 16:36:56 UTC; 12s ago
   Main PID: 718 (python3)
      Tasks: 5 (limit: 14131)
     Memory: 19.0M
        CPU: 218ms
     CGroup: /system.slice/telegram-bot.service
             `-718 /usr/bin/python3 /root/bot.py

Jan 12 16:36:56 ubuntu-server systemd[1]: Started Bot de Telegram para monitoreo y seguridad del servidor.
root@ubuntu-server:~#
```

Pruebas y Validación

Pruebas de conexión SSH:

Accede con un usuario normal (creado previamente empleado1 : empleado123).

Si accedemos por ssh, el bot nos avisará de que hay una nueva conexión entrante, debemos aceptarla o denegar manualmente

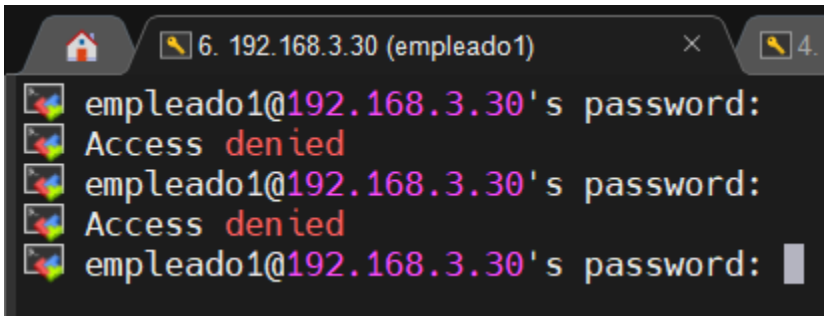


una vez dentro podremos ejecutar comandos libremente

```
empleado1@ubuntu-server:~$ whoami
empleado1
empleado1@ubuntu-server:~$
```

Accede con un usuario bloqueado y verifica el mensaje.

Vemos que si accedemos por ssh no se nos permitirá el acceso



Accede durante el modo seguro.

Vemos que si accedemos por ssh no se nos permitirá el acceso

```
empleado1@ubuntu-server:~$
Remote side unexpectedly closed network connection
```

Mantenimiento

Actualizar el bot: Edita el archivo bot.py y reinicia el servicio:

sudo systemctl restart telegram-bot

```
root@ubuntu-server:~# sudo systemctl restart telegram-bot
```

Revisar logs periódicamente:

sudo journalctl -u telegram-bot.service

```
root@ubuntu-server:~# sudo journalctl -u telegram-bot
```

Actualizar dependencias:

pip3 install --upgrade python-telegram-bot psutil pyTelegramBotAPI

```
root@ubuntu-server:~# pip install --upgrade python-telegram-bot psutil pyTelegramBotAPI
```