



Ejemplo Telnet en Python

Telnet es un protocolo de red utilizado para acceder a sistemas remotos a través de una conexión de texto, proporcionando una interfaz de línea de comandos para interactuar con un servidor. Aunque ha sido ampliamente reemplazado por protocolos más seguros como SSH, Telnet sigue siendo una herramienta valiosa para aprender sobre redes y administración remota.

La biblioteca socket en Python proporciona una interfaz para la comunicación de red utilizando el modelo de sockets. Esta biblioteca permite crear conexiones de red entre diferentes aplicaciones, ya sea en la misma máquina o en máquinas distintas a través de la red. En el contexto de Telnet, socket es fundamental porque permite implementar tanto servidores como clientes que se comunican usando este protocolo.

La biblioteca socket se utiliza para establecer un servidor Telnet que escucha conexiones de clientes y un cliente Telnet que se conecta al servidor. El servidor y el cliente intercambian mensajes a través de sockets, permitiendo la autenticación básica y el envío de datos entre ellos. La biblioteca socket facilita la creación de estas conexiones y la transmisión de datos en el formato que Telnet espera, gestionando detalles como la conexión de red y el intercambio de información en texto claro.

Para este ejemplo se hará un cliente y un servidor en Python para poder autenticarse, el servidor se va a encargar de recibir las credenciales del cliente y el servidor hará la validación de esas credenciales, si las credenciales son correctas mostrará un mensaje de Autenticación exitosa en el cliente y si es fallida mostrará el mensaje de Autenticación fallida.

Para ello se agrega la siguiente lógica del lado del cliente_telnet.py:

```
import socket

# Configuración del cliente
HOST = ip_del_servidor' # Reemplaza con la IP de tu servidor Ubuntu
PORT = 5000 # Debe coincidir con el puerto configurado en el servidor

# Conectar al servidor
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((HOST, PORT))

# Recibir el mensaje de bienvenida
print(client_socket.recv(1024).decode('utf-8'))

# Enviar nombre de usuario
username = input("Ingrese su nombre de usuario: ")
client_socket.sendall(username.encode('utf-8') + b"\n")

# Recibir la solicitud de contraseña
print(client_socket.recv(1024).decode('utf-8'))

# Enviar la contraseña
password = input("Ingrese su contraseña: ")
client_socket.sendall(password.encode('utf-8') + b"\n")

# Recibir el resultado de la autenticación
print(client_socket.recv(1024).decode('utf-8'))
```



```
client_socket.close()
```

Para el servidor_telnet.py se va a agregar la siguiente lógica para poder recibir las credenciales y autenticas:

```
import socket

# Configuración del servidor
HOST = '0.0.0.0' # Escuchar en todas las interfaces de red disponibles
PORT = 5000      # Puerto arbitrario que no requiere permisos de superusuario

# Crear el socket del servidor
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((HOST, PORT))
server_socket.listen(1)
print(f"Servidor Telnet escuchando en {HOST}:{PORT}...")

while True:
    client_socket, client_address = server_socket.accept()
    print(f"Conexión desde {client_address}")

    client_socket.sendall("Bienvenido al servidor Telnet!\nLogin: ".encode('utf-8'))

    # Recibir nombre de usuario
    username = client_socket.recv(1024).strip().decode('utf-8')

    # Pedir contraseña
    client_socket.sendall("Password: ".encode('utf-8'))
    password = client_socket.recv(1024).strip().decode('utf-8')

    # Autenticación simple
    if username == 'escom' and password == 'cisco':
        client_socket.sendall("\nAutenticación exitosa!\n".encode('utf-8'))
    else:
        client_socket.sendall("\nAutenticación fallida!\n".encode('utf-8'))

    client_socket.close()
```

Al ejecutar y realizar las pruebas se puede ver el siguiente mensaje del lado del cliente para el caso de autenticación exitosa:

```
python cliente_telnet.py
Bienvenido al servidor Telnet!
Login:
Ingrese su nombre de usuario: escom
Password:
Ingrese su contraseña: cisco

Autenticación exitosa!
```



Para el caso de la autenticación fallida se puede ver el siguiente mensaje:

```
python cliente_telnet.py
Bienvenido al servidor Telnet!
Login:
Ingrese su nombre de usuario: escom
Password:
Ingrese su contraseña: 1234

Autenticación fallida!
```

ACTIVIDAD:

- Realizar el mismo procedimiento de autenticación, pero usando la biblioteca telnetlib.