```python
import socket, platform, os

SRV_ADDR = ""
SRV_PORT = 1234

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((SRV_ADDR, SRV_PORT))
s.listen(1)
connection, address = s.accept()

print("client connected: ", address)

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

    if(data.decode('utf-8') == '1'):
        tosend = platform.platform() + " " + platform.machine()
        connection.sendall(tosend.encode())
    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"
        connection.sendall(tosend.encode())
    elif(data.decode('utf-8')== '0'):
        connection.close()
        connection, address = s.accept()
```

CONSEGNA [SETTIMANA 3 LEZIONE 4]
BACKDOOR

Il codice mostrato a fianco fa riferimento ad una *backdoor*, vediamo come funziona. Per *backdoor* si fa riferimento ad un accesso segreto in un Sistema informatico che permette di evitare procedure di autenticazione/sicurezza. Le *backdoor* possono essere create e usate legittimamente, ma nel caso venissero utilizzate per scopi illeciti prendono il nome di *remote access trojan (RAT).*

#IMPORTAZIONE DEI MODULI

Alla *riga 1* possiamo notare l'importazione di alcuni moduli come ***socket*** (consente l'accesso a comandi utili per la comunicazione di rete), ***platform*** (consente l'accesso ad informazioni riguardanti il terminale, come per esempio l'architettura) e ***os*** (consente accesso a funzionalità del Sistema operative, come la creazione di file e cartelle).

#CREAZIONE DEL SOCKET E CONNESSIONE ALLA RETE

Dalla *riga 3 alla riga 11* possiamo notare come il programma stia definendo alcune variabili (**SRV_ADDR** per l'indirizzo IP e **SRV_PORT** per la porta).
Alla riga 6 viene creato il socket definendo il tipo di connettività (ovvero **AF_INET** definisce **IPV4** e **SOCK_STREAM** il protocollo **TCP**).
Alla riga 7 vengono assegnati IP e porta al socket e viene messo in ascolto *alla riga 8*.
La riga 9 del codice serve per accettare le connessioni in entrata dai client e *alla riga 10* viene stampato a schermo il suo indirizzo IP.

#DECODIFICA DEI DATI E INTERCETTAZIONE DELLE INFORMAZIONI

Alla riga 13 inizia il **ciclo while**, in questo caso senza un **break()** per terminarlo. Viene poi inserito un costrutto **try-except**, dove *alla riga 15* il server stabilisce una connessione di 1024 byte continui e nel caso ci fossero delle eccezioni si passerebbe al ciclo successivo.
Dalla *riga 18* il server decodifica i dati ricevuti utilizzando l'encoding **UTF-8** e se il dato inserito fosse uguale a 1 allora verrebbero inviate informazioni sulla piattaforma in uso (tramite **platform.platform**) e sulla macchina (tramite **platform.machine**).
Se il dato ricevuto fosse uguale a 2 allora verrebbero ricevuti ulteriori dati dal client riguardo le cartelle (con **os.listdir**) e tramite il costrutto try-except si otterrebbe una lista di file presenti (con il **ciclo for**) per poi essere inviata. Nel caso ci fossero eccezioni dovute ad un percorso non valido entrerebbe in azione l'except, inviando *"wrong path"*. Infine se il dato ricevuto fosse uguale a 0 si terminerebbe la connessione lasciando il server nuovamente in ascolto e pronto per accettare nuove connessioni.