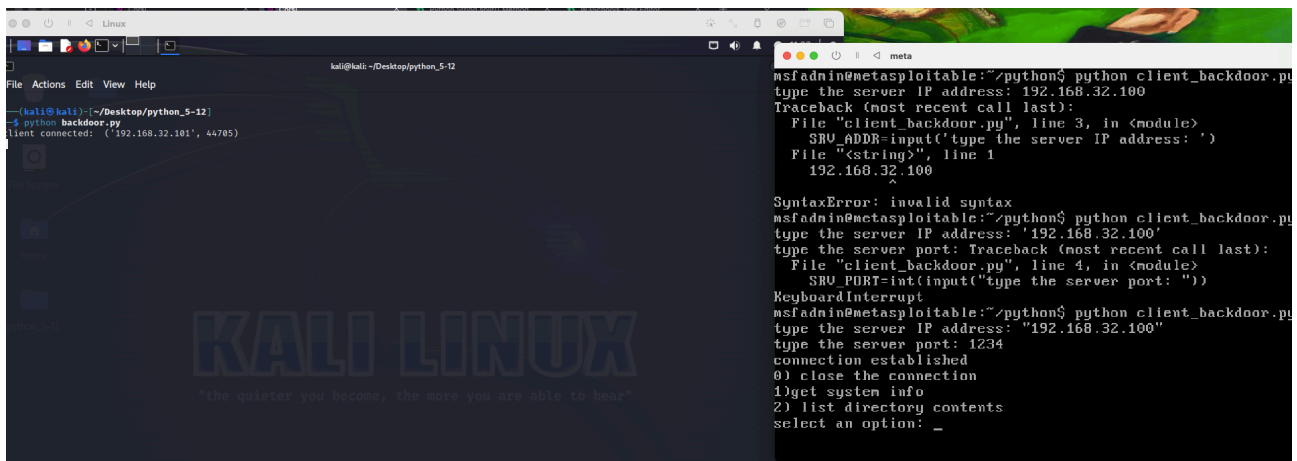


Una backdoor è una porta di accesso a un sistema informatico che consente a un utente remoto di controllarlo, fornisce un modo alternativo per entrare in un sistema senza passare attraverso le procedure di sicurezza convenzionali.

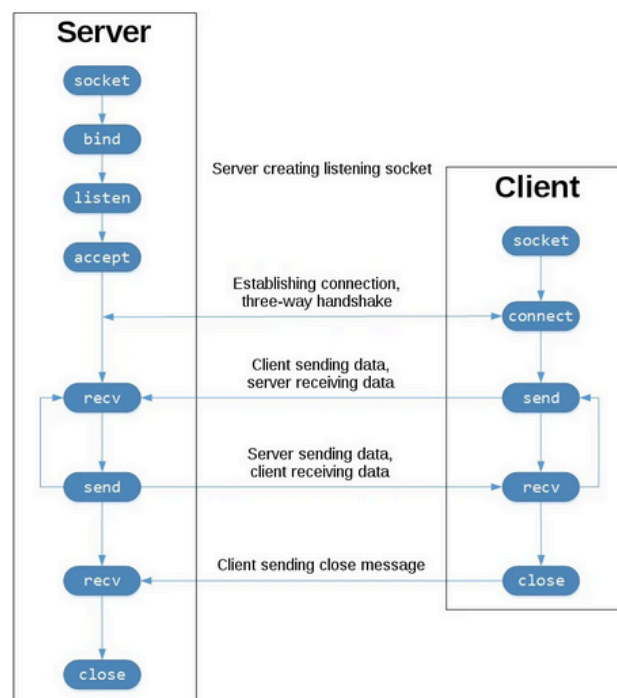
In questo caso importiamo la libreria socket per poter scambiare pacchetti(dati) tra due computer sulla stessa rete, prendiamo in considerazione una comunicazione di tipo client/server ascoltando le comunicazioni TCP in entrata su uno specifico indirizzo IP e porta.



```
msfadmin@metasploitable:~/python$ python client_backdoor.py
type the server IP address: 192.168.32.100
Traceback (most recent call last):
  File "client_backdoor.py", line 3, in <module>
    SRV_ADDR=input('type the server IP address: ')
  File "<string>", line 1
    192.168.32.100
    ^
SyntaxError: invalid syntax
msfadmin@metasploitable:~/python$ python client_backdoor.py
type the server IP address: '192.168.32.100'
type the server port: Traceback (most recent call last):
  File "client_backdoor.py", line 4, in <module>
    SRV_PORT=int(input("type the server port: "))
KeyboardInterrupt
msfadmin@metasploitable:~/python$ python client_backdoor.py
type the server IP address: "192.168.32.100"
type the server port: 1234
connection established
0) close the connection
1) get system info
2) list directory contents
select an option: _
```

Nella backdoor sorgente con il comando `socket.socket` creiamo una nuova socket specificando la tipologia di indirizzo IPV4, con il comando `s.bind` associamo un indirizzo IP ad una porta per poi con `s.listen` metterci in ascolto specificando il numero di connessioni massime in coda.

Quando il programma leggerà `s.accept`, la backdoor del server attenderà di stabilire una connessione con il client attraverso il comando `.connect()`, questo permetterà di avviare il three-way handshake.

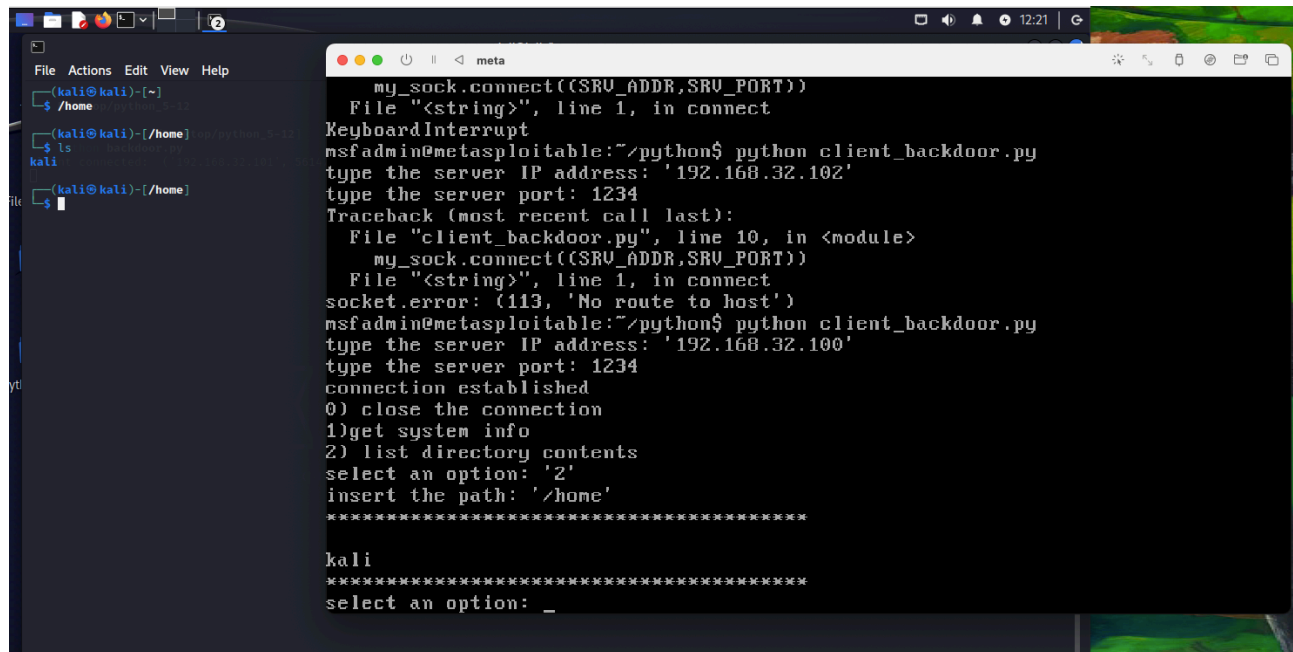


Il client dopo aver stabilito la connessione con il server ha tre opzioni:

0—>chiudere la connessione con `mysock.close()`

1—>richiamare la backdoor nel server per ottenere informazioni sul sistema operativo

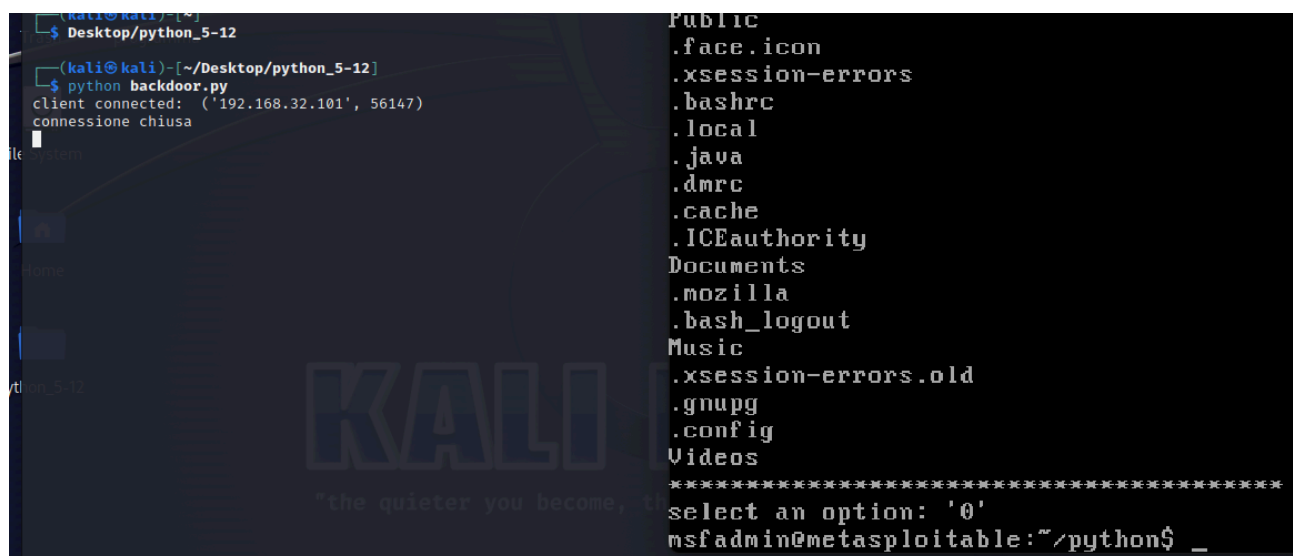
2—>inviare alla backdoor del server il path selezionato dall'utente in modo tale che venga restituito al client la lista effettiva di tutti gli elementi presenti nella directory scelta presente nel server.



```
(kali@kali)-[~]
$ /home
(kali@kali)-[/home]
$ ls
kali
(kali@kali)-[/home]
$

my_sock.connect((SRV_ADDR,SRV_PORT))
File "<string>", line 1, in connect
KeyboardInterrupt
msfadmin@metasploitable:~/python$ python client_backdoor.py
type the server IP address: '192.168.32.102'
type the server port: 1234
Traceback (most recent call last):
  File "client_backdoor.py", line 10, in <module>
    my_sock.connect((SRV_ADDR,SRV_PORT))
  File "<string>", line 1, in connect
socket.error: (113, 'No route to host')
msfadmin@metasploitable:~/python$ python client_backdoor.py
type the server IP address: '192.168.32.100'
type the server port: 1234
connection established
0) close the connection
1) get system info
2) list directory contents
select an option: '2'
insert the path: '/home'
*****

kali
*****
select an option: _
```



```
(kali@kali)-[~]
$ Desktop/python_5-12
(kali@kali)-[~/Desktop/python_5-12]
$ python backdoor.py
client connected: ('192.168.32.101', 56147)
connessione chiusa
Public
.face.icon
.xsession-errors
.bashrc
.local
.java
.dmrc
.cache
.ICEauthority
Documents
.mozilla
.bash_logout
Music
.xsession-errors.old
.gnupg
.config
Videos
*****
select an option: '0'
msfadmin@metasploitable:~/python$ _
```