

```
File "/usr/lib/python3.11/socket.py", line 294, in accept
    fd, addr = self._accept()
KeyboardInterrupt

(kali@kali)-[~/pythonsample]
$ python test.py

CTraceback (most recent call last):
  File "/home/kali/pythonsample/test.py", line 9, in <module>
    connection, address = s.accept()
                          ^^^^^^^^^
  File "/usr/lib/python3.11/socket.py", line 294, in accept
    fd, addr = self._accept()
KeyboardInterrupt

(kali@kali)-[~/pythonsample]
$ python test.py
Traceback (most recent call last):
  File "/home/kali/pythonsample/test.py", line 7, in <module>
    s.bind((SRV_ADDR, SRV_PORT))
  OSError: [Errno 98] Address already in use

(kali@kali)-[~/pythonsample]
$ python test.py
CTraceback (most recent call last):
  File "/home/kali/pythonsample/test.py", line 9, in <module>
    connection, address = s.accept()
                          ^^^^^^^^^
  File "/usr/lib/python3.11/socket.py", line 294, in accept
    fd, addr = self._accept()
KeyboardInterrupt

(kali@kali)-[~/pythonsample]
$ n
n: command not found

(kali@kali)-[~/pythonsample]
$ nano test.py

(kali@kali)-[~/pythonsample]
$ python test.py
```

```
File Actions Edit View Help
$ nano test2.py

(kali@kali)-[~/pythonsample]
$ nano test2.py

(kali@kali)-[~/pythonsample]
$ python test2.py
CTraceback (most recent call last):
  File "/home/kali/pythonsample/test2.py", line 9, in <module>
    connection, address = s.accept()
                          ^^^^^^^^^
  File "/usr/lib/python3.11/socket.py", line 294, in accept
    fd, addr = self._accept()
KeyboardInterrupt

(kali@kali)-[~/pythonsample]
$ python test2.py
Traceback (most recent call last):
  File "/home/kali/pythonsample/test2.py", line 7, in <module>
    s.bind((SRV_ADDR, SRV_PORT))
  OSError: [Errno 98] Address already in use

(kali@kali)-[~/pythonsample]
$ python test2.py
```

La differenza principale tra i due codici è che il primo rappresenta il lato del server che accetta le connessioni e gestisce le richieste dei client, mentre il secondo rappresenta il lato del client che stabilisce la connessione con il server e invia richieste specifiche al server. Il server è progettato per ascoltare le connessioni e rispondere alle richieste, mentre il client si connette al server e invia richieste per ottenere informazioni o eseguire azioni specifiche sul server.

Una backdoor è una vulnerabilità intenzionale o un meccanismo nascosto inserito all'interno di un sistema informatico, software o dispositivo da un utente malevolo. Essenzialmente, funge da accesso non autorizzato che permette a chi lo ha creato di accedere al sistema senza passare attraverso le normali procedure di autenticazione o sicurezza. La pericolosità delle backdoor risiede nel fatto che permettono ad attaccanti di accedere in modo non autorizzato ai sistemi, consentendo loro di rubare dati sensibili, danneggiare o controllare il sistema, distribuire malware o eseguire altre azioni dannose senza essere rilevati.