



EPICODE

Progetto Alessandro Moscetti 10/10



Una backdoor (letteralmente 'porta sul retro') è una porta d'accesso secondaria a un sistema informatico. La pericolosità delle backdoor sta nella loro capacità di bypassare le procedure di autenticazione. Possono, però, essere usate anche in modo legittimo, per esempio dai programmatori in caso di manutenzione.

```

SRV_ADDR = input("Type the server IP address: ")
SRV_PORT = int(input("Type the server port: "))

def print_menu():
    print("""\n\n0) Close the connection
1) Get system info
2) List directory contents""")

my_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
my_sock.connect((SRV_ADDR, SRV_PORT))

print("Connection established")
print_menu()

while 1:
    message = input("\n-Select an option: ")

    if(message == "0"):
        my_sock.sendall(message.encode())
        my_sock.close()
        break

    elif(message == "1"):
        my_sock.sendall(message.encode())
        data = my_sock.recv(1024)
        if not data: break
        print(data.decode('utf-8'))

    elif(message == "2"):
        path = input("Insert the path: ")
        my_sock.sendall(message.encode())
        my_sock.sendall(path.encode())
        data = my_sock.recv(1024)
        data = data.decode('utf-8').split(",")
        print("""40)
for x in data:
    print(x)
print("""40)

```

Il primo codice si collega come client a un server che inseriamo da input.

A seguire mostra un menù con le seguenti funzioni:

- '0'- Termina la connessione e termina il programma.
- '1'- Ricevere informazioni sul server.
- '2'- Inserire da input un percorso e ricevere informazioni specifiche.

```

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()

```

Il secondo programma crea un server a cui assegnamo indirizzo e porta da input e risponde alle richieste del client del precedente programma.

Grazie per l'attenzione.