Report Port Scanner

Il port scanning è una tecnica utilizzata per individuare i servizi in ascolto su una determinata macchina o rete, identificando le porte aperte e chiuse. Per individuare le librerie Python adatte al port scanning, sono state eseguite diverse fasi di ricerche online effettuando un ampio giro su motori di ricerca come Google e su siti di sviluppatori come GitHub, alla ricerca di librerie Python specificamente progettate per il port scanning.

Abbiamo studiato le principali librerie Python e sono state esaminate per valutare se offrivano funzionalità di port scanning o se potevano essere utilizzate per implementare tale funzionalità e dopo un'attenta ricerca e valutazione delle opzioni disponibili, abbiamo individuate le librerie più funzionali secondo le nostre conoscenze per il port scanning.

Successivamente il codice prosegue con un blocco di tipo try in modo che il programma non si bloccherà, ma gestirà l'errore in modo controllato.

Dopodiché parte un loop di tipo for che legge una sequenza di numeri di di porta da “porta\_iniziale” a “porta\_finale”

All'interno del loop, viene creato un oggetto socket (s) usando socket.socket(socket.AF\_INET, socket.SOCK\_STREAM), questo crea un socket di tipo TCP/IP.

socket.setdefaulttimeout(0.5) imposta un timeout di 0.5 secondi per la connessione. Ciò significa che se non riesce a connettersi al socket entro 0.5 secondi, passerà alla porta successiva.

s.connect\_ex((target,port)) tenta di stabilire una connessione al target specificato sulla porta specificata. connect\_ex restituirà 0 se la connessione ha successo, altrimenti restituirà un codice di errore corrispondente al tipo di errore riscontrato.

Se la connessione avrà successo verrà stampato un messaggio indicando che la porta è aperta.

Indipendentemente dall'esito della connessione, il socket viene chiuso con s.close() per rilasciare le risorse.