# Progetto Gestione di Reti A.A. 2024/2025

Nome: Diego

Cognome: Milletti

E-mail: d.milletti@studenti.unipi.it

## Introduzione

Il presente documento descrive il funzionamento di un progetto sviluppato per il rilevamento di anomalie nella dimensione della finestra TCP nei flussi di rete. Lo strumento, realizzato in linguaggio C, analizza file ".pcap" ed è in grado di individuare comportamenti anomali legati alla trasmissione TCP. I risultati delle analisi possono essere visualizzati direttamente sul terminale oppure esportati in un report testuale in formato ".txt".

## Prerequisiti

- Aver installato il compilatore **gcc (sudo apt-get install gcc)**
- Aver installato la libreria **ndpi (sudo apt-get install libpcap-dev)**
- Aver installato la libreria **libpcap (sudo apt-get install libpcap-dev)**

## Descrizione

Lo scopo principale del programma è analizzare i flussi TCP in modo **bidirezionale**, monitorando in particolare la **dimensione della finestra TCP** in ciascuna direzione. A tal fine sono state definite specifiche strutture dati:

- **FlowKey**: struttura utilizzata per identificare univocamente un flusso TCP. Contiene gli indirizzi IP e le porte di origine e destinazione, oltre a un intero is_ipv6 che permette di distinguere tra flussi IPv4 e IPv6.
- **FlowData**: struttura che rappresenta un intero flusso TCP. Include un FlowKey, un contatore di pacchetti (packet_count), un contatore delle finestre TCP pari a zero (zero_window_count), una struttura win_stats per raccogliere i valori assunti dalla finestra TCP durante la comunicazione, e un campo skipped_initial che tiene traccia del numero di pacchetti iniziali ignorati.
  In particolare, vengono **saltati i primi 3 pacchetti** del flusso (tipicamente appartenenti alla fase di handshake TCP a 3 vie) per evitare che **valori non significativi o stabilizzazioni iniziali** influenzino negativamente l'analisi statistica della finestra.

Il progetto è stato sviluppato a partire dal programma pcount.c, integrando diverse funzionalità provenienti dal software **nDPI** al fine di ottimizzare l'uso della memoria. In particolare, è stato utilizzato il puntatore "struct ndpi_analyze_struct *win_stats" per memorizzare i valori della finestra TCP per ogni pacchetto. Questo ha permesso di calcolare alcune statistiche significative, come **il coefficiente di variazione**, utile per l'individuazione di alcune anomalie.

Tra i criteri di rilevamento, il programma segnala anche la presenza di **finestre TCP di dimensione uguale a zero**. Se almeno un'occorrenza viene rilevata durante l'analisi, verrà generato un avviso, visualizzato sul terminale oppure scritto nel report in formato ".txt" (se è stato specificato il parametro "-r").

Per una corretta gestione dei pacchetti e flussi TCP, è stata modificata la funzione "dummyprocesssPacket" e sono state aggiunte funzioni ausiliari come:

- **Comparekeys**: confronta due strutture “Flowkey” per verificare se appartengono allo stesso flusso (nella stessa direzione).
- **Get_or_create_flow**: crea dinamicamente un nuovo flusso se non ancora presente, oppure restituisce il flusso esistente alla “Flowkey” in ingresso.
- **Cleanup_flows**: libera tutta la memoria allocata dinamicamente.

Infine, per eseguire l’analisi vera e propria, è stata implementata la funzione **flow_analysis**, che esamina la dimensione della finestra TCP per ciascun flusso. La funzione rileva eventuali anomalie, sia in modalità **normale** che **verbose**, e in quest’ultima fornisce anche ulteriori informazioni diagnostiche per i flussi non anomali, come il numero totale di pacchetti, la dimensione minima, massima e media della finestra TCP, oltre al coefficiente di variazione.

Come accennato in precedenza, per la segnalazione delle anomalie viene utilizzato il **coefficiente di variazione**.
Dopo diverse prove empiriche, utilizzando **Wireshark** e studiando il grafico della dimensione della finestra TCP di vari flussi, ho individuato delle **soglie dinamiche**, utili per evitare falsi positivi e garantire un’analisi più coerente con il comportamento reale dei flussi TCP, **tenendo conto che durante l’analisi non viene considerato il TCP Window Scaling**.

# Istruzioni per l’uso

1. **Compilazione**:
   Utilizzare il comando **make** per compilare automaticamente il codice.

2. **Esecuzione**:
   Per eseguire il programma in **modalità normale** (per visualizzare solo flussi anomali) usare il comando:
   “**./pcount -i <file.pcap> ”**
   Il filtro TCP verrà applicato di default.
   Per analizzare grandi file “.pcap”, è consigliato salvare l’output su un file report, aggiungendo il parametro “**-r <file_report.txt>”.**

   Invece per eseguire il programma in **modalità verbose** (visualizzando tutti i flussi, anche quelli non anomali) utilizzare il comando**:**
   **“./pcount -i <file.pcap> -v 1”**

3. **Pulizia**:
   Utilizzare il comando **make clean** per rimuovere i file eseguibili generati.

# Test

Ho effettuato vari test per controllare il corretto funzionamento del programma:

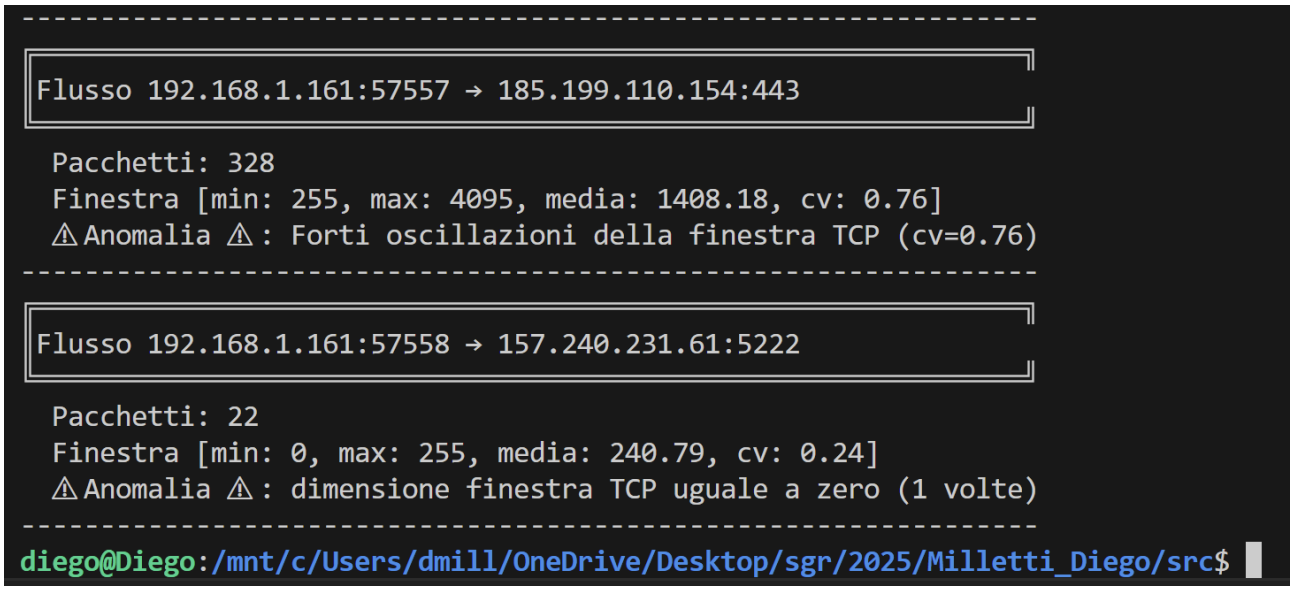
- È stato utilizzato il tool **Valgrind** per controllare l’eventuale presenza di perdite di memoria. Lo screenshot riportato di seguito mostra che l’unica perdita rilevata è dovuta a una funzione interna della libreria **libpcap**, non dipendente dal programma sviluppato.

```
==69388==
==69388== HEAP SUMMARY:
==69388==     in use at exit: 96 bytes in 1 blocks
==69388==   total heap usage: 39 allocs, 38 frees, 28,712 bytes allocated
==69388==
==69388== 96 bytes in 1 blocks are definitely lost in loss record 1 of 1
==69388==    at 0x484DA83: calloc (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
==69388==    by 0x48895BD: ??? (in /usr/lib/x86_64-linux-gnu/libpcap.so.1.10.1)
==69388==    by 0x487BB93: pcap_compile (in /usr/lib/x86_64-linux-gnu/libpcap.so.1.10.1)
==69388==    by 0x10CD6D: main (pcount.c:788)
==69388==
==69388== LEAK SUMMARY:
==69388==    definitely lost: 96 bytes in 1 blocks
==69388==    indirectly lost: 0 bytes in 0 blocks
==69388==      possibly lost: 0 bytes in 0 blocks
==69388==    still reachable: 0 bytes in 0 blocks
==69388==         suppressed: 0 bytes in 0 blocks
==69388==
==69388== For lists of detected and suppressed errors, rerun with: -s
==69388== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
diego@Diego:/mnt/c/Users/dmill/OneDrive/Desktop/sgr/2025/Milletti_Diego/src$
```

- Lo screenshot seguente rappresenta l'analisi senza anomalie di un flusso TCP presente nel file **443-safari.pcap** (incluso nel progetto)

```
======= ANALISI FLUSSI TCP =======
╔═══════════════════════════════════════════════════════════╗
Flusso 192.168.1.13:53031 → 178.62.197.130:443
╚═══════════════════════════════════════════════════════════╝
  Pacchetti: 21
  Finestra [min: 3890, max: 4096, media: 4065.50, cv: 0.01]
---------------------------------------------------------------
╔═══════════════════════════════════════════════════════════╗
Flusso 178.62.197.130:443 → 192.168.1.13:53031
╚═══════════════════════════════════════════════════════════╝
  Pacchetti: 20
  Finestra [min: 506, max: 508, media: 506.59, cv: 0.00]
---------------------------------------------------------------
======= Nessuna anomalia rilevata =======
diego@Diego:/mnt/c/Users/dmill/OneDrive/Desktop/sgr/2025/Milletti_Diego/src$
```

- Nel file **firenze1.pcapng** (anch'esso incluso), l'analisi ha correttamente individuato le seguenti anomalie: è stata rilevata una finestra TCP con dimensione uguale a zero e una forte oscillazione, come riportato dal **coefficiente di variazione**.




- Questo screenshot mostra l'analisi di un flusso IPv6 contenuto nel file **http_ipv6.pcap** (anch'esso allegato). Nel secondo flusso sono stati analizzati meno di 10 pacchetti, motivo per cui il programma segnala che i dati non sono sufficienti per un'analisi affidabile. Ho ritenuto utile stampare comunque queste informazioni per evidenziare i limiti dei dati disponibili e mantenere la trasparenza nell'output.