

## **Lab-Redes T1 - Monitoramento de Tráfego em Túnel**

**Brenda Pereira Camara, João Pedro Salles da Silva, Leonardo Bertoletti**

### **1. Introdução**

Este relatório descreve o desenvolvimento e uso de um sistema para monitoramento de tráfego de rede em ambientes virtualizados, utilizando Java e C. O sistema permite criar um túnel de rede entre dois dispositivos (cliente e servidor) e monitorar, em tempo real, os pacotes que trafegam por esse túnel, registrando informações detalhadas em arquivos CSV para análise posterior.

### **2. Estrutura Geral do Sistema**

O sistema é composto por dois principais componentes:

#### **Túnel de Rede (traffic\_tunnel):**

Implementado em C, cria uma interface virtual (tun0) em cada máquina, conectando cliente e servidor. O tráfego enviado para tun0 em uma ponta é transmitido para a outra ponta, simulando uma VPN ponto-a-ponto.

#### **Monitor de Tráfego (Java):**

Um programa em Java que utiliza a biblioteca Pcap4J para capturar pacotes em tempo real de qualquer interface de rede (física ou virtual). O monitor exibe estatísticas na tela e salva detalhes dos pacotes em arquivos CSV, separados por camada (enlace, rede e transporte).

### **3. Requisitos e Evidências de Implementação**

#### **Criação e uso do túnel de rede:**

O túnel é iniciado com comandos específicos em cada máquina, usando os IPs corretos das interfaces físicas.

#### **Exemplo de comando no cliente:**

```
./traffic_tunnel eth0 -c 172.31.66.2 -t ./client1.sh &
```

#### **Exemplo de comando no servidor:**

```
./traffic_tunnel eth0 -s 172.31.66.1 -t ./server.sh &
```

Após a configuração, cada máquina possui uma interface tun0 ativa.

#### **Monitoramento do tráfego**

O monitor é executado com o comando:

```
java -jar target/network-monitor-1.0.0.jar tun0
```

Ele exibe estatísticas como número de pacotes, bytes e protocolos detectados.

Os dados são salvos em arquivos CSV (camada2.csv, camada3.csv, camada4.csv).

#### **Geração e captura de tráfego**

Para testar, comandos como ping são usados para gerar tráfego entre as interfaces tun0 dos dois lados. O monitor registra o tráfego gerado, comprovando o funcionamento do sistema.

#### **Evidências**

Os arquivos CSV mostram os pacotes capturados, com informações como endereços IP, portas, protocolos e tamanhos.

O monitor exibe em tempo real o aumento dos contadores de pacotes e bytes ao gerar tráfego pelo túnel.

```
MONITOR DE TRAFEGO DE REDE
Interface: tun0

2025-06-23 02:59:01

=====
ESTATISTICAS GERAIS
=====
Total de Pacotes: 3
Total de Bytes: 0
Taxa Media: 0.30

=====
PROTOCOLOS DE REDE
=====
IPv6      : 3
=====
PROTOCOLOS DE TRANSPORTE
=====
Other     : 3
=====
Pressione Ctrl+C para parar o monitor...

[DEBUG] Read tun device
60 00 00 00 00 08 3a ff fe 80 00 00 00 00 00 00
1c be 06 b9 7a 06 ca ec ff 02 00 00 00 00 00 00
00 00 00 00 00 00 02 85 00 14 cd 00 00 00 00 00
[DEBUG] Sent packet
[DEBUG] Read tun device
60 00 00 00 00 08 3a ff fe 80 00 00 00 00 00 00
1c be 06 b9 7a 06 ca ec ff 02 00 00 00 00 00 00
00 00 00 00 00 00 02 85 00 14 cd 00 00 00 00 00
[DEBUG] Sent packet

root@41e674bcalc2:/home/lab-redes-project# ping google.com
ping: google.com: Temporary failure in name resolution
root@41e674bcalc2:/home/lab-redes-project# ping -c 4 172.31.66.1
PING 172.31.66.1 (172.31.66.1) 56(84) bytes of data:
64 bytes from 172.31.66.1: icmp_seq=1 ttl=64 time=0.252 ms
64 bytes from 172.31.66.1: icmp_seq=2 ttl=64 time=0.553 ms
64 bytes from 172.31.66.1: icmp_seq=3 ttl=64 time=0.302 ms
64 bytes from 172.31.66.1: icmp_seq=4 ttl=64 time=0.199 ms

--- 172.31.66.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 0.199/0.326/0.553/0.135 ms

[DEBUG] Sent packet
[DEBUG] Read tun device
45 00 00 3a 33 1e 40 00 40 11 0d 05 ac 1f 42 65
08 08 04 04 91 c2 00 35 00 26 5c 6c 9b 9c 01 00
00 01 00 00 00 00 00 00 08 66 61 63 65 62 6f 6f
6b 03 63 6f 6d 00 00 01 00 01
[DEBUG] Sent packet
[DEBUG] Read tun device
45 00 00 3a 33 1f 40 00 40 11 0d 04 ac 1f 42 65
08 08 04 04 91 c2 00 35 00 26 53 4f a4 9e 01 00
00 01 00 00 00 00 00 00 08 66 61 63 65 62 6f 6f
6b 03 63 6f 6d 00 00 1c 00 01
[DEBUG] Sent packet

root@41e674bcalc2:/home/lab-redes-project# ping google.com
ping: google.com: Temporary failure in name resolution
root@41e674bcalc2:/home/lab-redes-project# ping -c 4 172.31.66.1
PING 172.31.66.1 (172.31.66.1) 56(84) bytes of data:
64 bytes from 172.31.66.1: icmp_seq=1 ttl=64 time=0.252 ms
64 bytes from 172.31.66.1: icmp_seq=2 ttl=64 time=0.553 ms
64 bytes from 172.31.66.1: icmp_seq=3 ttl=64 time=0.302 ms
64 bytes from 172.31.66.1: icmp_seq=4 ttl=64 time=0.199 ms

--- 172.31.66.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 0.199/0.326/0.553/0.135 ms
root@41e674bcalc2:/home/lab-redes-project# ping google.com
ping: google.com: Temporary failure in name resolution
root@41e674bcalc2:/home/lab-redes-project# facebook.com
bash: facebook.com: command not found
```

## 4. Considerações Finais

O sistema desenvolvido permitiu criar um ambiente de rede virtualizado, onde foi possível monitorar detalhadamente o tráfego que passa por um túnel VPN customizado. Aprendemos a importância de configurar corretamente os IPs e interfaces para que o tráfego realmente passe pelo túnel, além de entender como funciona a captura de pacotes em diferentes camadas da rede. O uso de arquivos CSV facilitou a análise dos dados coletados. Enfrentamos desafios como garantir que o binário do túnel não estivesse travado e que o tráfego realmente passasse pela interface tun0, mas conseguimos superar esses obstáculos com testes e ajustes nos comandos. O trabalho foi fundamental para consolidar conhecimentos sobre tunelamento, monitoramento e análise de tráfego em redes.