

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**BÁO CÁO BÀI THỰC HÀNH 2**

**Kỹ thuật theo dõi và giám sát an toàn mạng**

**Lập trình với thư viện pcap**

**Giảng viên: Ninh Thị Thu Trang**

**Sinh viên: Hoàng Trung Kiên**

**Mã sinh viên: B20DCAT098**

**Hà Nội – 2024**

Mục lục

[1. Mục đích. 3](#_Toc162809152)

[2. Yêu cầu đối với sinh viên. 3](#_Toc162809153)

[3. Nội dung thực hành. 3](#_Toc162809154)

[3.1 Dấu vết không rõ ràng. 3](#_Toc162809155)

[3.2. Thống kê lưu lượng cơ bản. 4](#_Toc162809156)

[4. Checkwork. 8](#_Toc162809157)

# 1. Mục đích.

- Bài thực hành này giới thiệu về việc lập trình sử dụng thư viện/API pcap.

# 2. Yêu cầu đối với sinh viên.

- Sinh viên có hiểu biết cơ bản về dữ liệu gói tin đầy đủ, định dạng file pcap

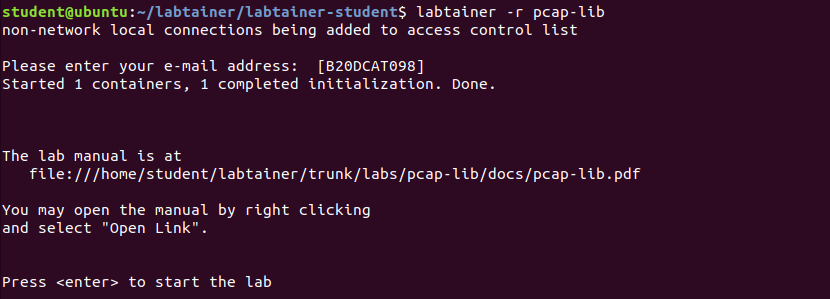
- Sinh viên phải có kiến thức cơ bản về dòng lệnh Linux.

- Sinh viên có kỹ năng lập trình cơ bản với C++ hoặc Python.

# 3. Nội dung thực hành.

Khởi động bài lab: Trên terminal, gõ lệnh:

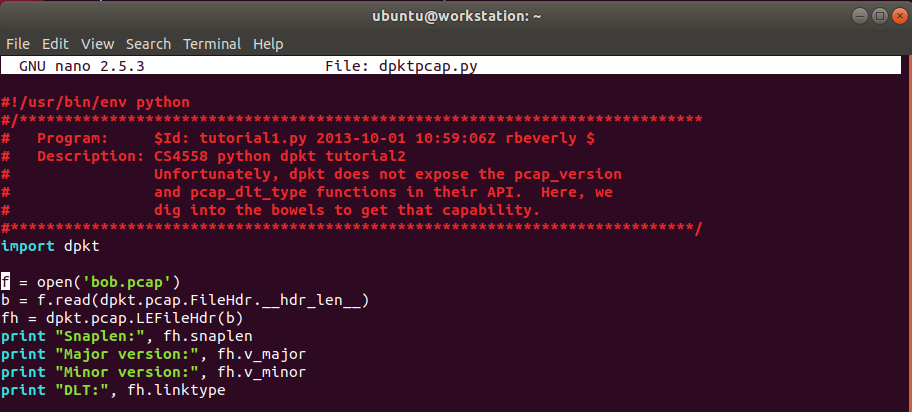
labtainer -r pcap-lib



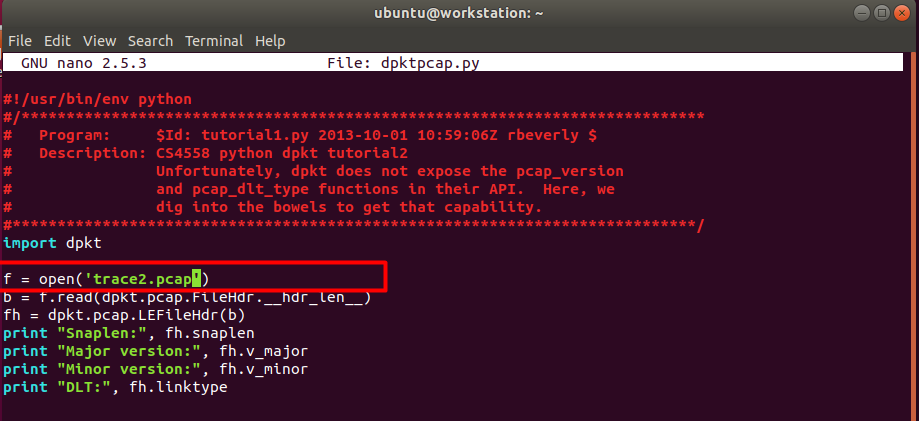
## 3.1 Dấu vết không rõ ràng.

Chúng ta sẽ phân tích một dấu vết gói tin ẩn danh lấy từ một điểm trao đổi Internet. Dấu vết này có trong thư mục gốc của bạn ở định dạng trace2.pcap.

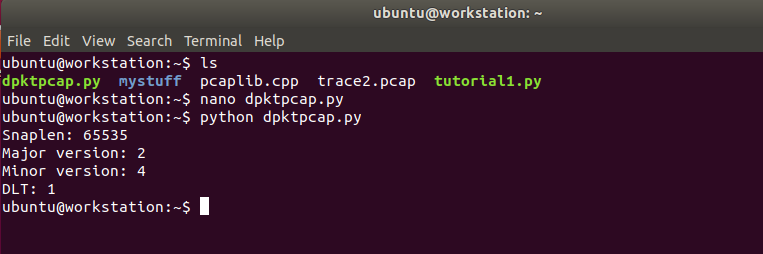
Mở file dpktpcap.py



Thay đổi file bob.pcap thành file trace2.pcap



=> sau đó chạy file dpktpcap.py



+ Loại liên kết trong dấu vết là: Giá trị DLT trong ảnh là 1, biểu thị liên kết Ethernet

+ Snaplen là viết tắt của snapshot length. Nó là một giá trị được sử dụng trong các công cụ ghi lại lưu lượng mạng để xác định số lượng byte được ghi lại từ mỗi gói tin.

+ Sự khác biệt giữa pcap và Pcapng:

Pcap:

•Cũ hơn, phổ biến hơn.

•Ít tính năng hơn, đơn giản hơn.

•Hiệu suất ghi/đọc nhanh hơn.

•Phù hợp cho ghi/phân tích lưu lượng cơ bản.

PcapNg:

•Mới hơn, ít phổ biến hơn.

•Nhiều tính năng hơn, linh hoạt hơn.

•Hiệu suất ghi/đọc chậm hơn.

•Phù hợp cho phân tích chuyên sâu, giải mã, lưu trữ lâu dài.

+ Tỉ lệ giao thức: ko có

## 3.2. Thống kê lưu lượng cơ bản.

Mở file tutorial1.py và viết đoạn code này vào

import dpkt

import socket

fd = open('trace2.pcap', 'rb')

pcap = dpkt.pcap.Reader(fd)

ipv4\_count = 0

non\_ipv4\_count = 0

first\_timestamp = None

last\_timestamp = None

packet\_count = 0

for ts, data in pcap:

if first\_timestamp is None:

first\_timestamp = ts

last\_timestamp = ts

eth = dpkt.ethernet.Ethernet(data)

if isinstance(eth.data, dpkt.ip.IP):

ipv4\_count += 1

else:

non\_ipv4\_count += 1

packet\_count += 1

#Print the results

print("IPv4 count: {}".format(ipv4\_count))

print("Non-IPv4 count: {}".format(non\_ipv4\_count))

#Print the first timestamp with at least two decimal places

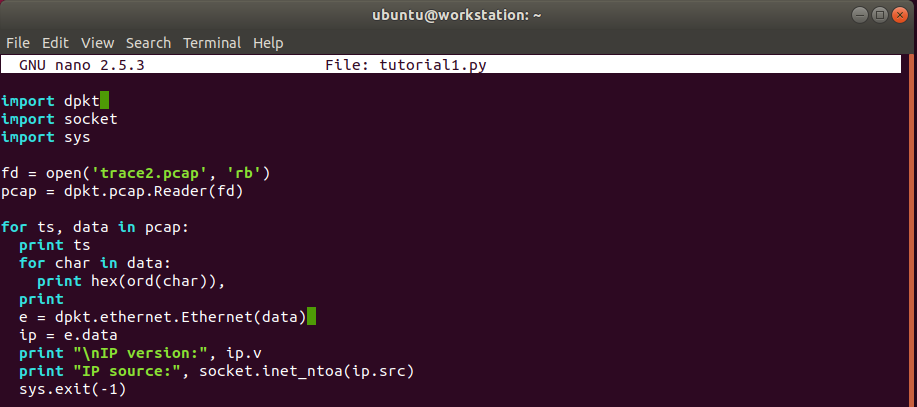
print("First timestamp:{:.2f}".format(first\_timestamp))

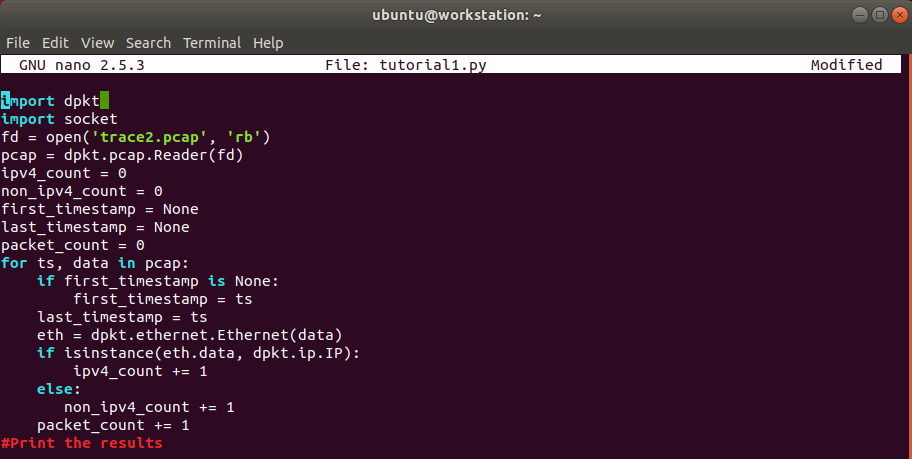
#Caculate and print the average packet rate

time\_span = last\_timestamp - first\_timestamp

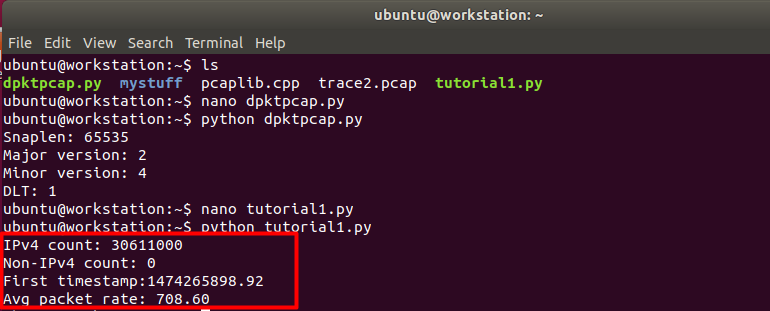
avg\_packet\_rate = packet\_count / time\_span if time\_span > 0 else 0

print("Avg packet rate: {:.2f}".format(avg\_packet\_rate))





Sau đó chạy file tutorial1.py



+ Dấu vết chứa 30611000 gói tin IPv4

+ Dấu vết chứa 0 gói tin không phải là IPv4

+ Dấu vết có timestamp của gói tin đầu tiên là 1474265898.92

+ Tốc độ trung bình của gói tin trong dấu vết là 708.60.

- Sau tiêu đề pcap cho mỗi gói tin là dữ liệu lưu lượng. Cải thiện chức năng gọi lại của bạn để giải mã các gói tin IP. Chức năng gọi lại của bạn nên thu được địa chỉ IP nguồn và đích, giao thức (ví dụ như TCP, UDP, ICMP, v.v.), cổng vận chuyển nguồn và đích (nếu có), v.v.

- Mở file tutorial1.py và sửa code

import dpkt

import socket

fd = open('trace2.pcap', 'rb')

pcap = dpkt.pcap.Reader(fd)

unique\_source\_addresses = set() # Set to store unique source addresses

unique\_dest\_addresses = set() # Set to store unique destination

source\_bytes = {} # Dictionary to store bytes sent by each source IP

source\_packet\_counts = {} # Dictionary to store packet counts sent by each source IP

for ts, data in pcap:

eth = dpkt.ethernet.Ethernet(data)

if isinstance(eth.data, dpkt.ip.IP):

ip = eth.data

# Convert bytes to string

if isinstance(ip.src, bytes):

src\_ip = socket.inet\_ntoa(ip.src)

unique\_source\_addresses.add(src\_ip)

if src\_ip in source\_bytes:

source\_bytes[src\_ip] += len(ip)

else:

source\_bytes[src\_ip] = len(ip)

if src\_ip in source\_packet\_counts:

source\_packet\_counts[src\_ip] += 1

else:

source\_packet\_counts[src\_ip] = 1

if isinstance(ip.dst, bytes):

dst\_ip = socket.inet\_ntoa(ip.dst)

unique\_dest\_addresses.add(dst\_ip)

print("Unique sources: {}".format(len(unique\_source\_addresses)))

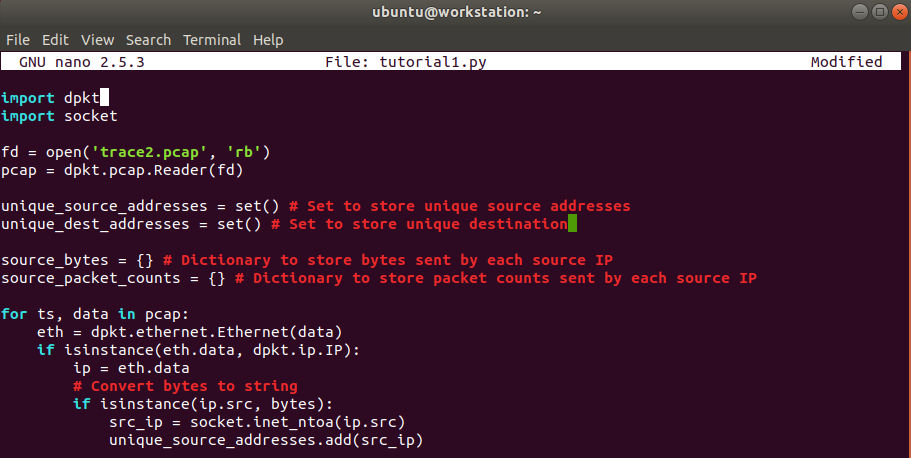
print("Unique destinations: {}".format(len(unique\_dest\_addresses)))

source\_with\_most\_bytes = max(source\_bytes, key=source\_bytes.get)

source\_with\_most\_packets = max(source\_packet\_counts, key=source\_packet\_counts.get)

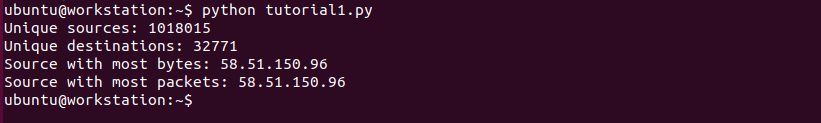
print("Source with most bytes: {}".format(source\_with\_most\_bytes))

print("Source with most packets: {}".format(source\_with\_most\_packets))



Sau đó chạy lại file tutorial1.py





+ Có bao nhiêu địa chỉ nguồn IPv4 duy nhất trong file dấu vết: 1018015

+ Có bao nhiêu địa chỉ đích IPv4 duy nhất trong dấu vết: 32771

+ IP nguồn nào đã gửi nhiều byte nhất: 58.51.150.96

+ IP nguồn nào đã gửi nhiều gói tin nhất: 58.51.150.96

- Dựa trên phân tích của dấu vết:

+ Liệt kê 3 đặc điểm không bình thường của lưu lượng mạng mà bạn thấy:

+ Tốc độ gói tin cao: Dấu vết cho thấy tốc độ gói tin trung bình là 708,60 gói tin mỗi giây. Đây là tốc độ cao hơn nhiều so với lưu lượng mạng thông thường, có thể là do một cuộc tấn công mạng hoặc hoạt động độc hại khác đang diễn ra.

+ Số lượng gói tin thấp: Mặc dù tốc độ gói tin cao, nhưng số lượng gói tin tổng thể chỉ là 3.061.100. Điều này cho thấy rằng mỗi gói tin có kích thước rất nhỏ, có thể là do một cuộc tấn công quét mạng hoặc hoạt động dò tìm thông tin đang diễn ra.

+ Kích thước gói tin nhỏ: Kích thước gói tin trung bình không được hiển thị trong dấu vết, nhưng có thể là rất nhỏ dựa trên tốc độ gói tin cao và số lượng gói tin thấp. Kích thước gói tin nhỏ thường được sử dụng trong các cuộc tấn công mạng để tránh bị phát hiện.

+ Dựa trên các đặc điểm không bình thường được xác định, có thể đưa ra một số giải thích hợp lý về lưu lượng mạng mà dấu vết đại diện:

+ Cuộc tấn công từ chối dịch vụ (DoS): Một cuộc tấn công DoS có thể là nguyên nhân khiến tốc độ gói tin cao và số lượng gói tin thấp. Trong một cuộc tấn công DoS, kẻ tấn công sẽ gửi một lượng lớn lưu lượng truy cập đến một máy chủ hoặc mạng, khiến cho máy chủ hoặc mạng quá tải và không thể phục vụ các yêu cầu hợp pháp.

+ Cuộc tấn công quét mạng: Kích thước gói tin nhỏ có thể là dấu hiệu của một cuộc tấn công quét mạng. Trong một cuộc tấn công quét mạng, kẻ tấn công sẽ gửi các gói tin nhỏ đến nhiều địa chỉ IP khác nhau để tìm kiếm các máy tính dễ bị tấn công.

+ Hoạt động dò tìm thông tin: Kích thước gói tin nhỏ cũng có thể được sử dụng trong các hoạt động dò tìm thông tin. Kẻ tấn công có thể sử dụng các gói tin nhỏ để thăm dò mạng và xác định các dịch vụ đang chạy trên các máy tính.

# 4. Checkwork.

