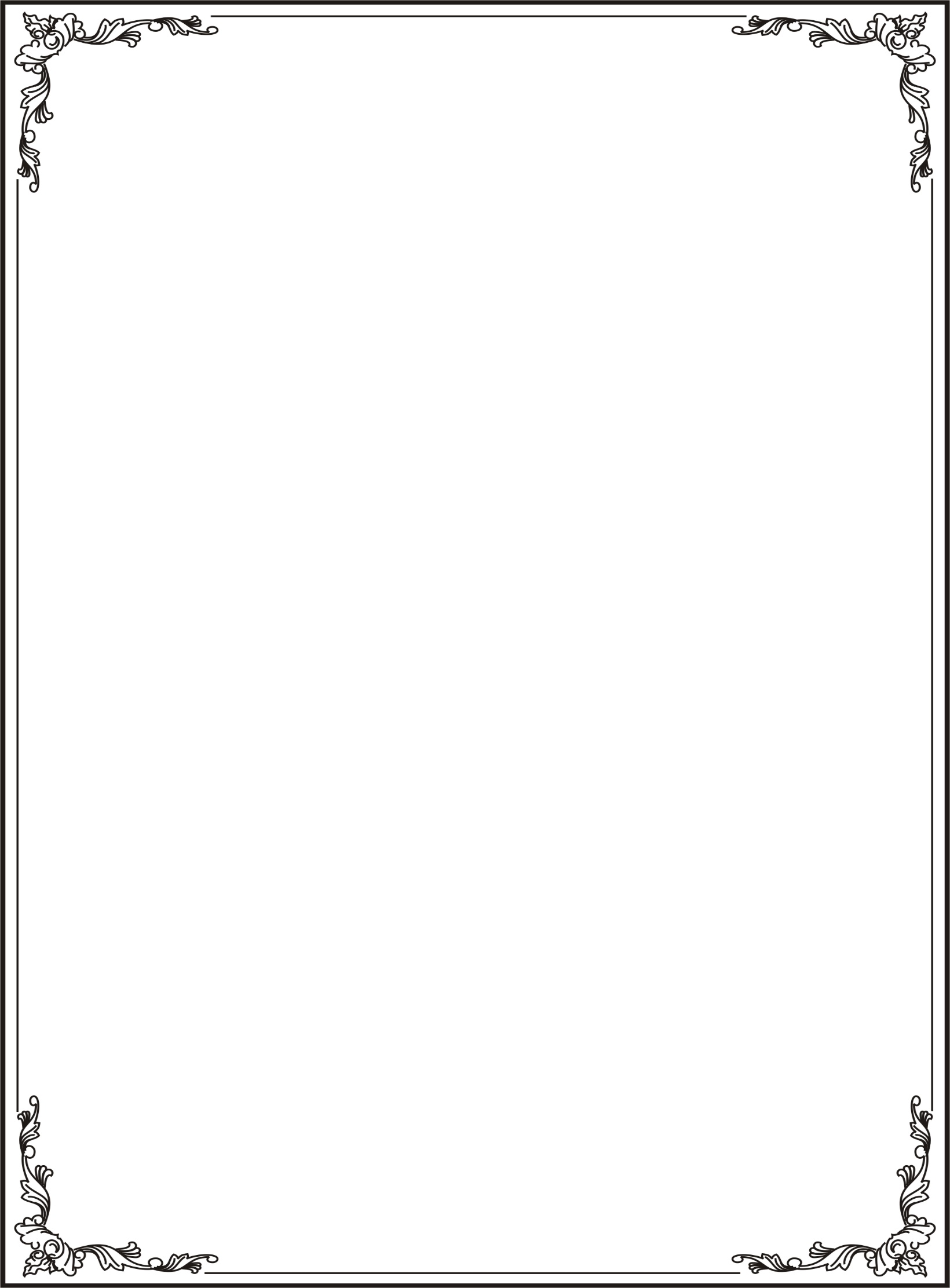
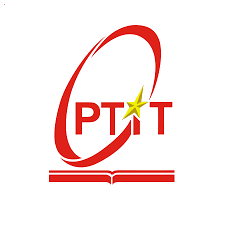
****

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO BÀI TẬP LỚN**

**MÔN CƠ SỞ AN TOÀN THÔNG TIN**

***Đề tài: Phân Tích và Kiểm Soát ứng dụng bằng kết nối DNS***

**Giảng viên hướng dẫn:** Đỗ Xuân Chợ

**Nhóm lớp:** 02

**Nhóm thực hiện:** 06

**Sinh viên tham gia:**

Hồ Thị Kiều Trinh – B21DCAT188

Phạm Thùy Trang – B21DCAT184

Nguyễn Thị Thanh Ngoan – B21DCAT145

Phan Hoàng Yến Nhi – B21DCAT148

**Hà Nội – 2023**

**MỤC LỤC**

[**I.** **GIỚI THIỆU** 3](#_Toc152573009)

[**II.** **LÝ THUYẾT** 3](#_Toc152573010)

[1. MongoDB và PyMongo 3](#_Toc152573011)

[2. Scapy 3](#_Toc152573012)

[3. Subprocess và Psutil 3](#_Toc152573013)

[**III.** **CÁC BƯỚC THỰC HIỆN** 3](#_Toc152573014)

[1. Thu thập domain và địa chỉ IP của các ứng dụng 3](#_Toc152573015)

[2. Đưa dữ liệu vào CSDL 6](#_Toc152573016)

[3. Viết mã nguồn Python để quản lý tiến trình dựa trên kết nối đến một địa chỉ IP hoặc tên miền. 7](#_Toc152573017)

[**IV.** **Kết quả** 10](#_Toc152573018)

1. **GIỚI THIỆU**

Chương trình theo dõi mạng được phát triển bằng Python sử dụng các thư viện như ***subprocess***, ***psutil***, ***pymongo***, và ***scapy***. Mục tiêu là theo dõi kết nối mạng, xác định ứng dụng tương ứng và chặn các tiến trình liên quan. Dưới đây là các phần lý thuyết liên quan và bước thực hiện chi tiết.

1. **LÝ THUYẾT**
2. **MongoDB và PyMongo**

* **MongoDB** là một hệ quản lý cơ sở dữ liệu Non – SQL dựa trên tài liệu JSON.
* **PyMongo** là một thư viện Python cho phép kết nối và tương tác với MongoDB từ chương trình Python.

1. **Scapy**

* **Scapy** là một thư viện Python mạnh mẽ cho việc tạo, gửi và nhận gói tin mạng.

Trong bài toán này, Scapy được sử dụng để bắt gói tin DNS.

1. **Subprocess và Psutil**

* Thư viện **subprocess** cho phép chương trình Python gọi các lệnh hệ thống khác.
* **Psutil** là một thư viện cho phép theo dõi thông tin hệ thống và quản lý tiến trình.

1. **CÁC BƯỚC THỰC HIỆN**
2. **Thu thập domain và địa chỉ IP của các ứng dụng**

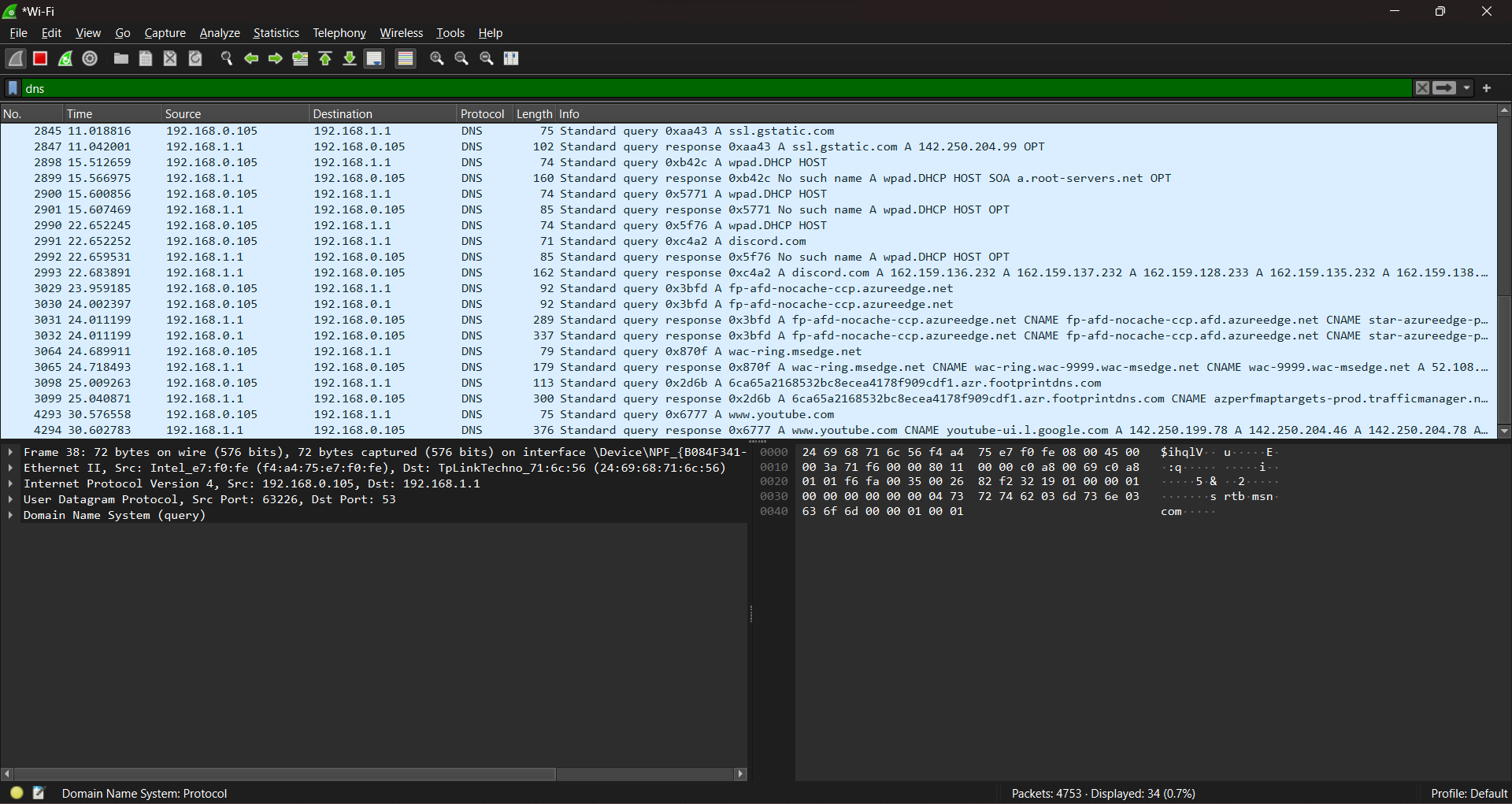
Để thu thập thông tin về domain và địa chỉ IP của các ứng dụng, có nhiều phương pháp khác nhau. Trong bài báo cáo này, sẽ trình bày hai cách phương pháp thực hiện như sau:

* ***Cách 1: Sử dụng công cụ WireShark:***

Công cụ WireShark là một bộ giải mã gói tin mạng mạnh mẽ, giúp theo dõi và phân tích gói tin trên mạng. Trong bài báo cáo này, chúng ta sẽ chỉ quan tâm đến gói tin DNS, vì trong đó chứa thông tin về domain và địa chỉ IP liên quan đến ứng dụng.

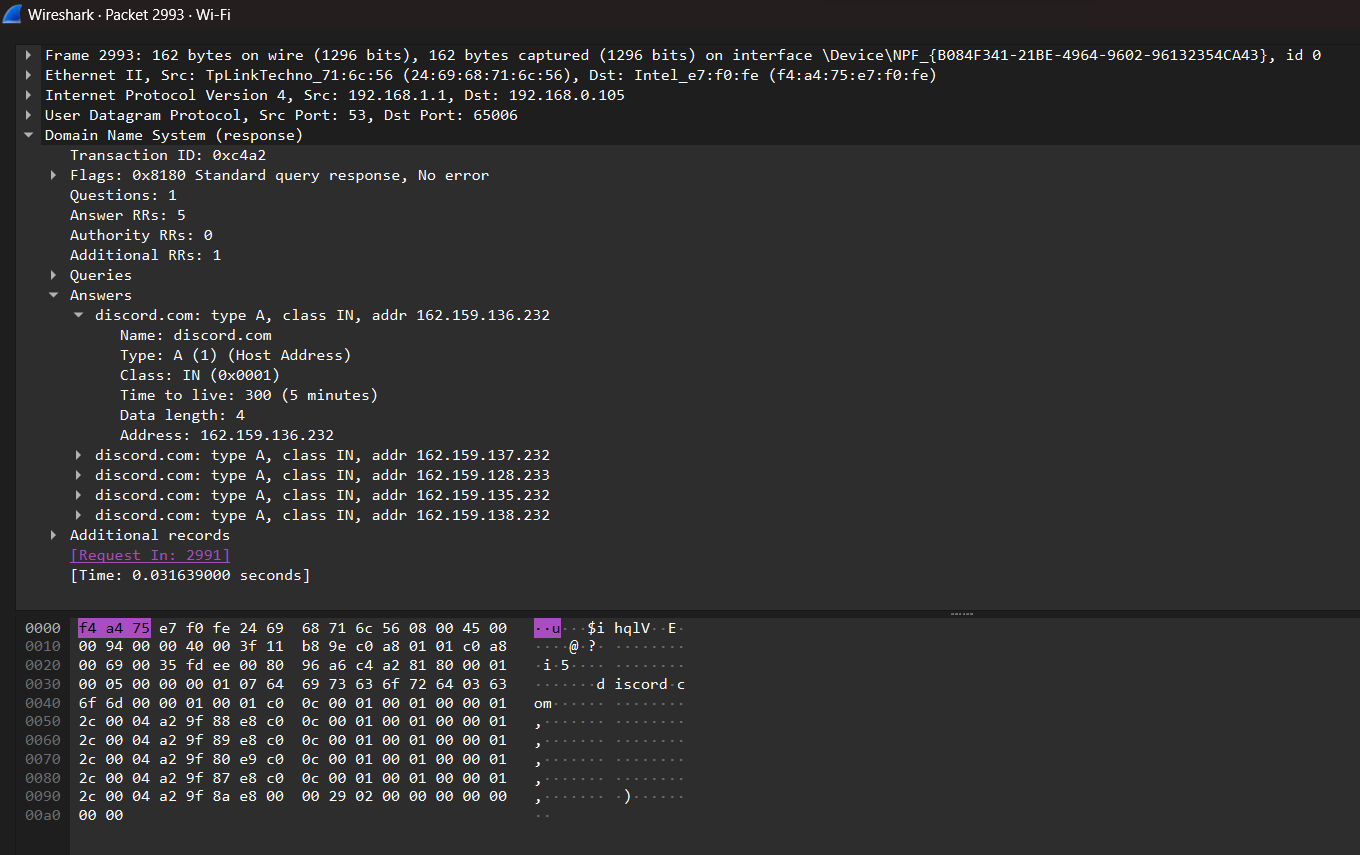
* **Bước 1:** Lọc Gói Tin DNS

Trước tiên, mở WireShark và chọn giao diện mạng cần theo dõi. Sau đó, sử dụng bộ lọc để chỉ hiển thị gói tin DNS:



* **Bước 2:** Quan Sát Thông Tin Domain và Địa Chỉ IP

Khi áp dụng bộ lọc, WireShark sẽ chỉ hiển thị các gói tin DNS. Trong các gói tin này, chúng ta có thể quan sát thông tin như tên miền (domain) được truy vấn và địa chỉ IP tương ứng.

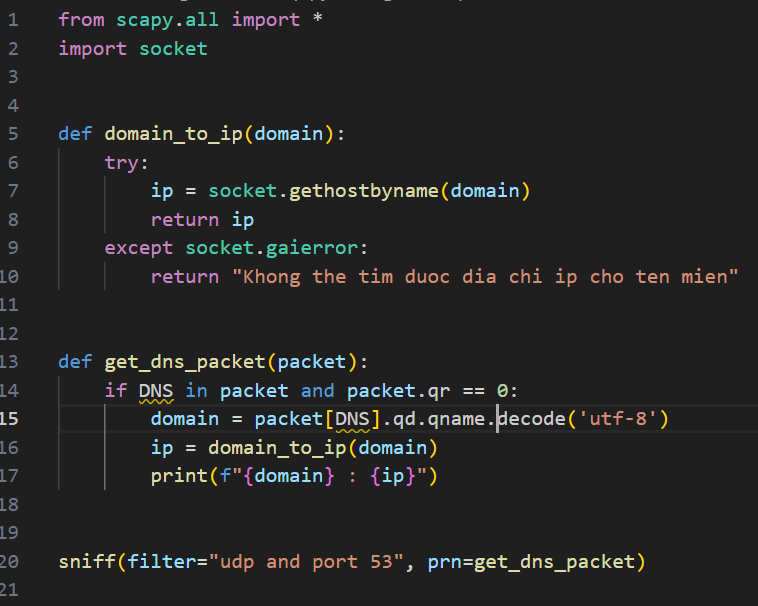


**KL:** Sử dụng công cụ giám sát mạng WireShark là một phương pháp hiệu quả để thu thập thông tin về domain và địa chỉ IP liên quan đến các ứng dụng trong môi trường mạng. Qua quá trình quan sát gói tin DNS, chúng ta có thể nhanh chóng xác định các hoạt động mạng và tìm ra các kết nối giữa tên miền và địa chỉ IP.

* ***Cách 2: Sử dụng thư viện Scapy:***

Thư viện Scapy của Python là một công cụ mạnh mẽ giúp thu thập thông tin mạng và thực hiện các thao tác mạng đa dạng. Trong báo cáo này, chúng ta sẽ tập trung vào việc sử dụng Scapy để thu thập thông tin về domain và địa chỉ IP của các ứng dụng.

Viết mã nguồn bằng Python



Mã nguồn Python trên sử dụng thư viện Scapy để lắng nghe gói tin trên mạng và in ra thông tin về các gói tin DNS query (gói tin yêu cầu thông tin về tên miền). Dưới đây là giải thích chi tiết về mã nguồn:

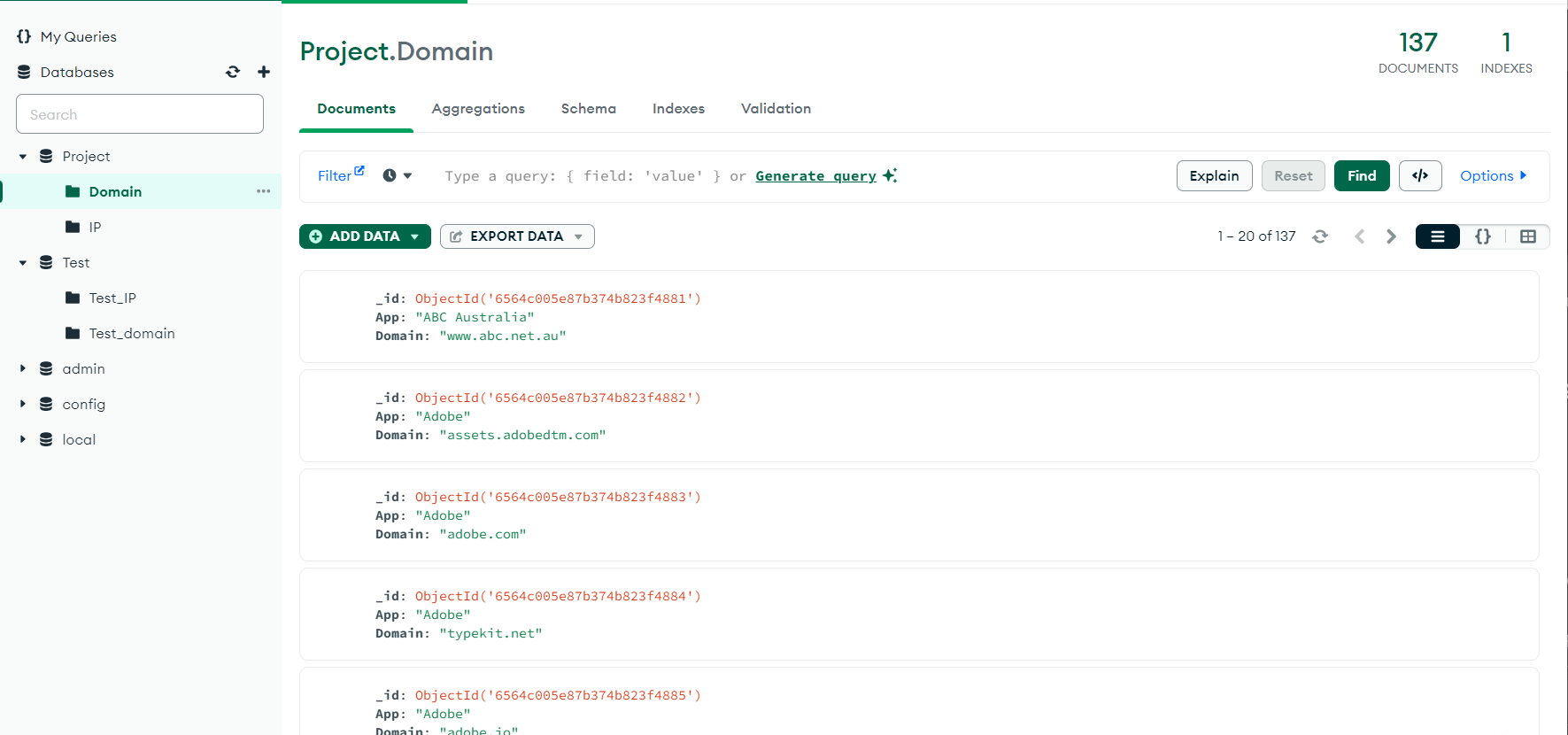
* Hàm ‘**domain\_to\_ip**’: Hàm này nhận vào một tên miền và chuyển đổi nó thnafh địa chỉ IP bằng các sử dụng hàm ‘gethostbyname’ của thư viện socket.
* Hàm ‘**get\_dns\_packet**’:
* Hàm này được sử dụng để xử lý gói tin DNS khi nó được bắt bởi Scapy.
* Kiểm tra xem gói tin có chứa lớp DNS (‘DNS in packet’) và có phải là gói tin DNS query không (‘packet.qr == 0’). Nếu đúng thì nó tricchs xuất tiên miền từ gói tin và chuyển đổi thành địa chỉ IP bằng gọi hàm ‘domain\_to\_ip’
* Cuối cùng, nó in ra thông tin về tên miền và địa chỉ IP.
* Sniffing gói tin DNS:
* Sử dụng hàm ‘**sniff**’ để lắng nghe gói tin trên mạng
* ‘**filter = “udp and port 53”**’: Chỉ lắng nghe các gói tiên UDP trên cổng 53, nơi thường được sử dụng cho DNS.
* ‘**prn = get\_dns\_packet**’: Khi một gói tin được bắt được, nó sẽ gọi hàm ‘get\_dns\_packet’ để xử lý gói tin đó.

**KL:** Sử dụng mã nguồn Python cũng là một cách hiệu quả để theo dõi và thu thập thông tin từ các gói tin DNS query trên mạng. Việc sử dụng Scapy và socket giúp chúng ta thực hiện các chức năng này một cách linh hoạt và mạnh mẽ, đồng thời mang lại khả năng ứng dụng rộng rãi trong lĩnh vực an ninh mạng và quản lý mạng.

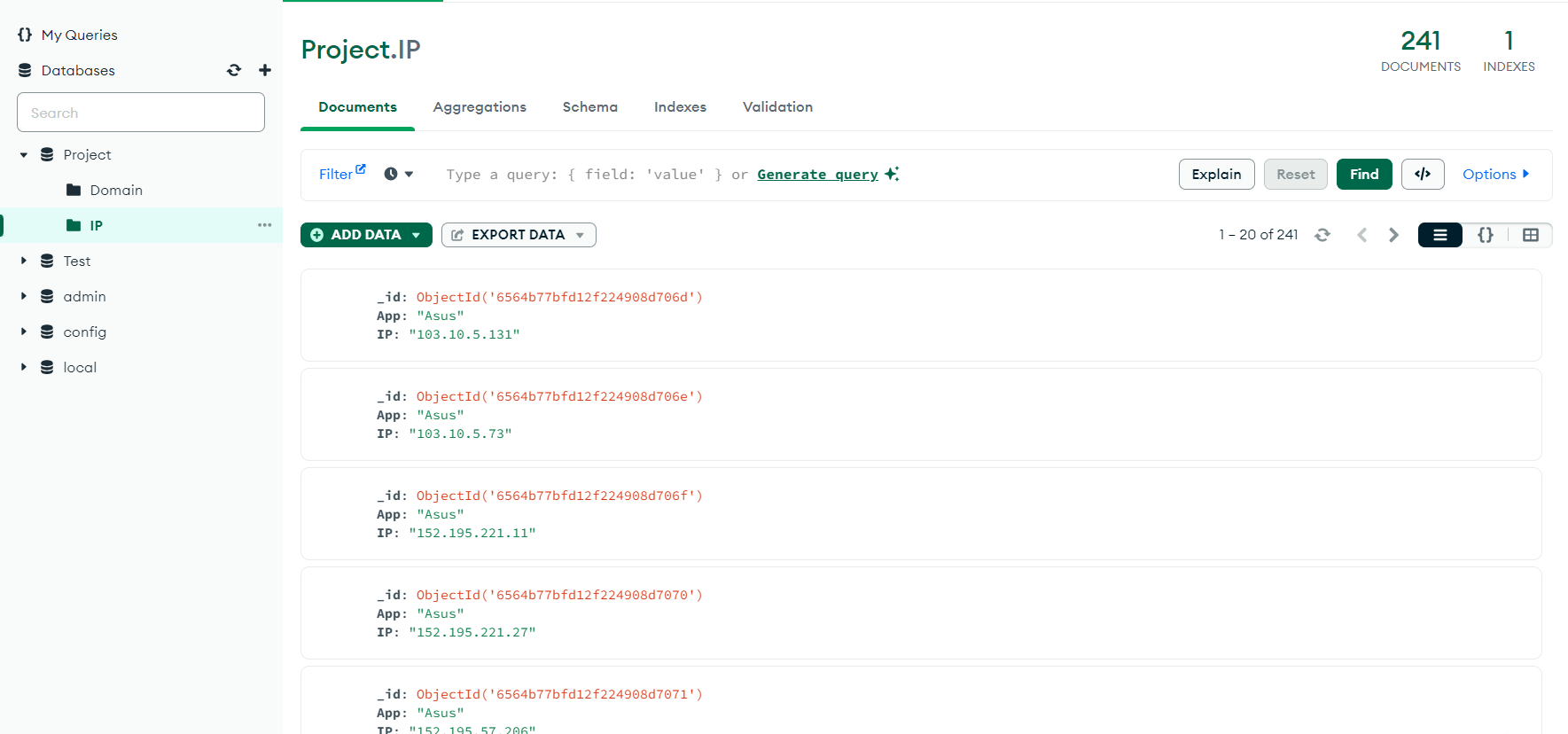
1. **Đưa dữ liệu vào CSDL**

Sử dụng hệ quản lý cơ sở dữ liệu MongoDB để lưu trữ thông tin. Tạo database “**Project**” với 3 collection là “**Domain**”, “**IP**” và “**Phan\_loai**” để chứa thông tin tương ứng về tên ứng dụng, domain, và địa chỉ IP.

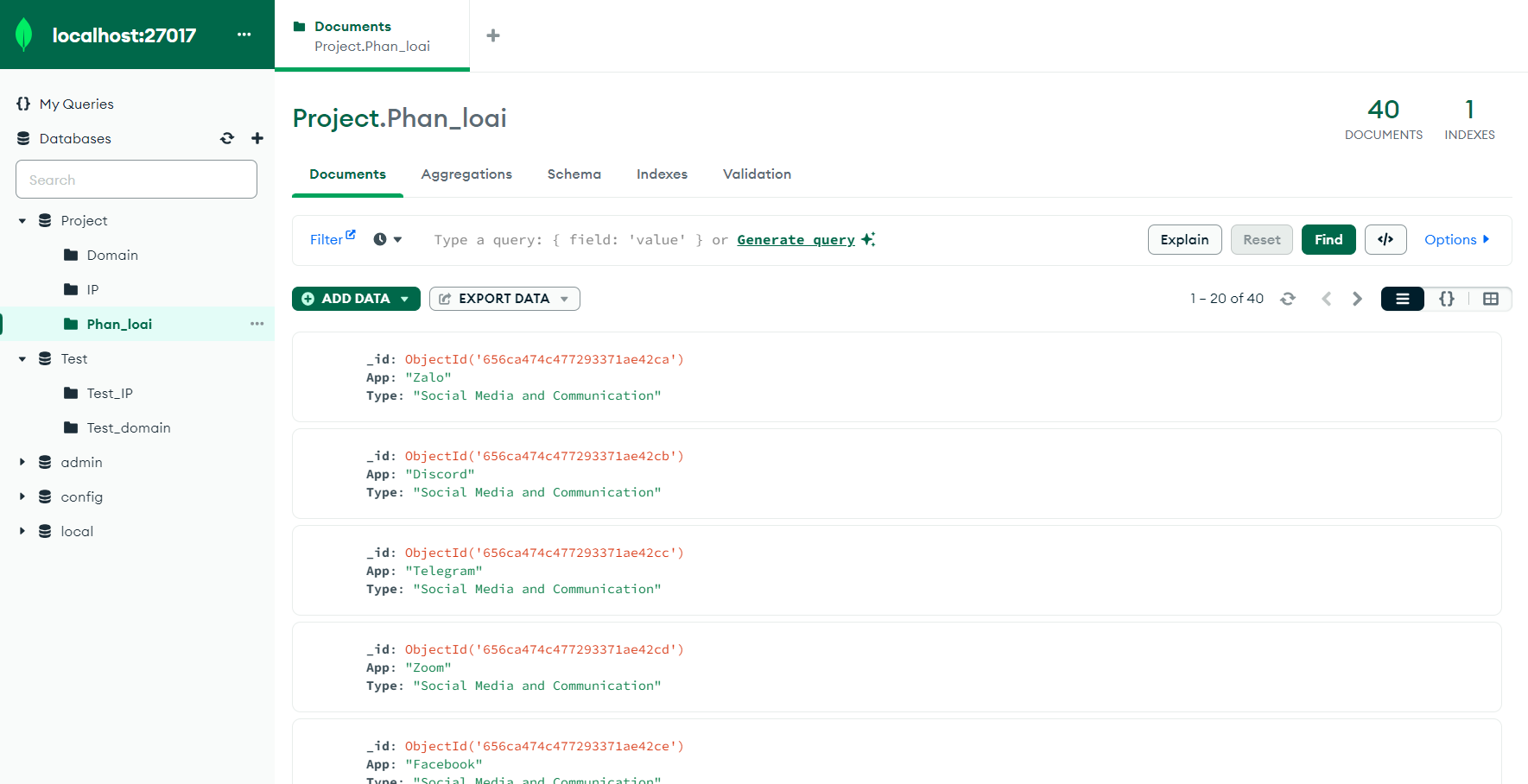
Collection “**Domain**” dùng để lưu thông tin về tên ứng dụng và domain tương ứng của nó.



Collection “**IP**” dùng để lưu thông tin về tên ứng dụng và các địa chỉ IP tương ứng của nó.



Collection “**Phan\_loai**” dùng để lưu thông tin về tên ứng dụng và phân loại.



1. **Viết mã nguồn Python để quản lý tiến trình dựa trên kết nối đến một địa chỉ IP hoặc tên miền.**

from datetime import datetime, time as datetime\_time

import psutil

from pymongo import MongoClient

from scapy.all import \*

import socket

client = MongoClient('mongodb://localhost:27017')

db = client['Test']

domain\_collection = db['Test\_domain']

ip\_collection = db['Test\_IP']

def domain\_correct(domain):

    parts = domain.split(".")

    result = ".".join(parts[-2:])

    if result.endswith("."):

        result = result[:-1]

    return result

def domain\_to\_app(domain):

    query = {'Domain': domain}

    result = domain\_collection.find\_one(query)

    if result:

        app = result.get('App')

        return app

    else:

        domain1 = domain\_correct(domain)

        query1 = {'Domain': domain1}

        result1 = domain\_collection.find\_one(query1)

        if result1:

            app1 = result1.get('App')

            return app1

        else:

            return "Không xác định ứng dụng"

def ip\_to\_app(ip\_address):

    query = {'IP': ip\_address}

    result = ip\_collection.find\_one(query)

    if result:

        app = result.get('App')

        return app

    else:

        return "Không xác định ứng dụng"

def domain\_to\_ip(domain\_name):

    try:

        ip\_address = socket.gethostbyname(domain\_name)

        return ip\_address

    except socket.gaierror:

        return "Không thể tìm thấy địa chỉ IP cho tên miền."

def is\_within\_allowed\_time():

    current\_time = datetime.now().time()

    start\_time = datetime\_time(7, 0)

    end\_time = datetime\_time(21, 0)

    return start\_time <= current\_time <= end\_time

def kill\_processes(App):

    try:

        all\_processes = psutil.process\_iter()

        App = App.lower()

        for process in all\_processes:

            try:

                if App in process.name().lower():

                    process.terminate()

                    print(f"Không cho phép truy cập {App} vào lúc này!!!")

            except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess):

                pass

    except Exception as e:

        print(f"Lỗi: {e}")

def dns\_packet\_handler(packet):

    if DNS in packet and packet.qr == 0:

        domain = packet[DNS].qd.qname.decode('utf-8')

        if domain.endswith("."):

            domain = domain[:-1]

        app = domain\_to\_app(domain)

        if app != "Không xác định ứng dụng":

            ip\_addr = domain\_to\_ip(domain)

            if (ip\_addr != "Không thể tìm thấy địa chỉ IP cho tên miền."):

                if (ip\_to\_app(ip\_addr) == "Không xác định ứng dụng"):

                    ip\_collection.insert\_one({'App': app, 'IP': ip\_addr})

            if is\_within\_allowed\_time():

                kill\_processes(app)

            print(f"{domain} : {app}")

        else:

            ip\_address = domain\_to\_ip(domain)

            if ip\_address != "Không thể tìm thấy địa chỉ IP cho tên miền.":

                app\_from\_ip = ip\_to\_app(ip\_address)

                if app\_from\_ip != "Không xác định ứng dụng":

                    domain\_collection.insert\_one(

                        {'App': app\_from\_ip, 'Domain': domain})

                    if is\_within\_allowed\_time():

                        kill\_processes(app\_from\_ip)

                    print(f"{domain} : {ip\_address} : {app\_from\_ip}")

                else:

                    print(f"{domain} : {ip\_address} : Không xác định ứng dụng")

            else:

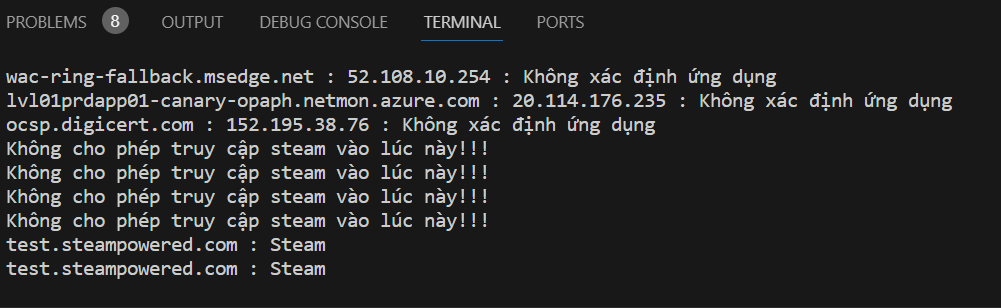
                print(f"{domain} : Không xác định ứng dụng")

# Bắt gói tin DNS

sniff(filter="udp and port 53", prn=dns\_packet\_handler)

1. **Kết quả**

Tiến trình của ứng dụng bị tắt đi trong khoảng thời gian không cho phép

****