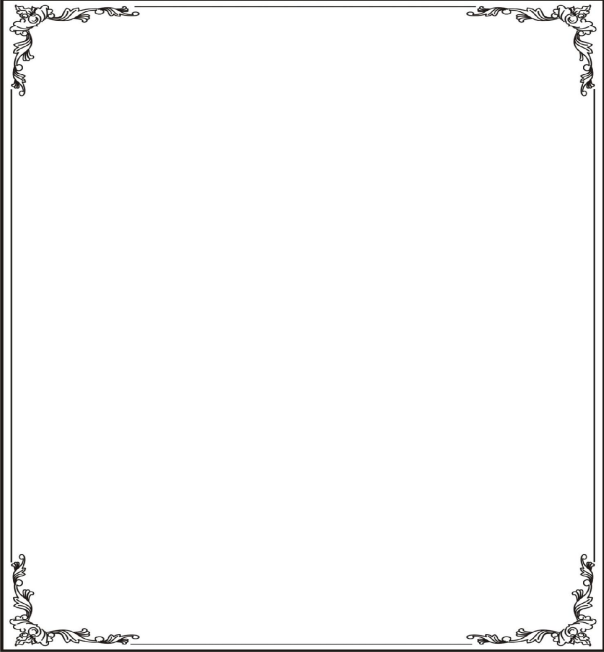
**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

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

**KHOA CÔNG NGHỆ THÔNG TIN 2**

🙟🕮🙝

Logo

Description automatically generated

**BÁO CÁO CUỐI KỲ**

**IoT VÀ ỨNG DỤNG**

**ĐỀ TÀI: TÌM HIỂU VỀ DẠNG TẤN CÔNG MẠNG TRONG MÔI TRƯỜNG IoT**

**NHÓM 9**

**Sinh viên thực hiện:**

1. **Trần Hữu Chiến N20DCCN008**
2. **Lê Văn Lương N20DCCN038**

***Giảng viên*: Đàm Minh Lịnh**

---oOo---

MỤC LỤC

[**I.** **Tìm hiểu về mô hình học máy Decision Tree** 3](#_Toc155338883)

[**1.** **Giới thiệu** 3](#_Toc155338884)

[**2.** **Thuật toán Classification and Regression Tree** 3](#_Toc155338885)

[**3.** **Độ thuần khiết** 4](#_Toc155338886)

[**II.** **Thực hiện hai dạng tấn công** 6](#_Toc155338887)

[**1.** **Tấn công SYN Flood** 6](#_Toc155338888)

[**1.1** **SYN Flood attack là gì** 6](#_Toc155338889)

[**1.2** **Các bước thực hiện tấn công SYN Flood** 6](#_Toc155338890)

[**1.3** **Thực hiện** 7](#_Toc155338891)

[**2.** **Tấn công ICMP Flood** 11](#_Toc155338892)

[**2.1** **Giới thiệu** 11](#_Toc155338893)

[**2.2** **Các bước thực hiện** 13](#_Toc155338894)

[**III.** **Huấn luyện bộ dữ liệu:** 17](#_Toc155338895)

[**1.** **Công nghệ sử dụng:** 17](#_Toc155338896)

[**2.** **Thực hiện huấn luyện bằng mô hình học máy Decision Tree** 17](#_Toc155338897)

[**2.1** **Xử lý dữ liệu** 17](#_Toc155338898)

[**IV.** **Kết luận** 24](#_Toc155338899)

1. **Tìm hiểu về mô hình học máy Decision Tree**
2. **Giới thiệu**

Mô hình cây quyết định (Decision Tree) là một phương pháp học máy có khả năng giải quyết cả hai loại vấn đề: phân loại (classification) và dự đoán (regression). Mục tiêu của cây quyết định là phân loại dữ liệu hoặc dự đoán giá trị đầu ra dựa trên các quyết định tại các nút của cây.

Việc xây dựng một decision tree trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng. Một điểm đáng lưu ý của decision tree là nó có thể làm việc với các đặc trưng (trong các tài liệu về decision tree, các đặc trưng thường được gọi là thuộc tính – attribute) dạng categorical, thường là rời rạc và không có thứ tự.

Một cây quyết định bắt đầu từ một nút gốc và phân chia dữ liệu thành các nhóm con thông qua các quy tắc quyết định, đặc trưng quan trọng được lựa chọn tại mỗi bước. Các nhánh và lá của cây đại diện cho các quyết định và kết quả cuối cùng. Quá trình xây dựng cây được thực hiện dựa trên các thuật toán như ID3, C4.5, CART, và Random Forest.

Một ưu điểm lớn của Decision Tree là tính diễn giải cao, cho phép người dùng dễ dàng hiểu cách mô hình ra quyết định. Tuy nhiên, cần chú ý đến việc kiểm soát quá mức khớp, và các kỹ thuật như pruning và điều chỉnh tham số là quan trọng để đảm bảo hiệu suất và tổng quát hóa của mô hình.

Mô hình Decision Tree đã được ứng dụng rộng rãi trong nhiều lĩnh vực như phân loại email, dự đoán bệnh tật, và các ứng dụng khác nơi yêu cầu tính minh bạch và khả năng giải thích của mô hình.

1. **Thuật toán Classification and Regression Tree**

Thuật toán CART (cây phân loại và hồi quy - Classification and Regression Tree), là một thuật toán được Leo Breiman giới thiệu năm 1984, dùng để xây dựng cây quyết định đa năng, với mục đích phân loại và hồi quy dữ liệu.

Thuật toán CART là nền tảng cho những thuật toán khác mạnh mẽ hơn như bagged decision trees (tập hợp nhiều cây quyết định để cùng đoán nhận), random forest (rừng ngẫu nhiên) và boosted decision trees (tạp hợp các cây khác nhau, là các model; và một siêu model - meta model - để tập hợp kết quả đoán nhận của các model con)

Biểu diễn của mô hình CART chính là một cây nhị phân. Cây nhị phân này cũng chính là cây nhị phân thường thấy trong môn cấu trúc dữ liệu và giải thuật, với mỗi nút là một biến độc lập riêng lẻ x (trong học máy, ta gọi đây là các đặc trưng), và điểm tách (hay là ngưỡng phân loại) của đặc trưng đó. Nút lá của cây chứa biến đầu ra y , dùng để thực hiện đoán nhận.

Cách thuật toán CART hoạt động:

* Chia Nhánh (Splitting): Bắt đầu từ nút gốc, thuật toán chọn một đặc trưng và giá trị cắt (threshold) để chia nhánh tập dữ liệu thành hai phần. Mục tiêu là chia nhánh sao cho tăng độ tinh khiết của dữ liệu. Đối với bài toán phân loại, thông tin Gini (Gini impurity) thường được sử dụng để đo độ tinh khiết, trong khi đối với bài toán hồi quy, Mean Squared Error (MSE) thường được sử dụng.
* Dừng Lại (Stopping Criteria): Quá trình chia nhánh tiếp tục cho đến khi một số điều kiện dừng được đáp ứng. Các điều kiện này có thể bao gồm đạt đến độ sâu tối đa (max\_depth), số lượng mẫu tối thiểu tại mỗi lá (min\_samples\_leaf), hoặc một điều kiện khác tùy thuộc vào yêu cầu cụ thể.
* Tính Diễn Giải: Cây quyết định được tạo ra có thể được hiểu và giải thích dễ dàng. Mỗi nút trong cây đại diện cho một quyết định dựa trên giá trị của một đặc trưng, và các lá đại diện cho kết quả dự đoán.
* Pruning (Tắt Nhánh): Tắt nhánh (pruning) là quá trình loại bỏ các nhánh không quan trọng để kiểm soát quá mức khớp và tối ưu hóa hiệu suất của mô hình. Một số phương pháp pruning như "Cost-Complexity Pruning" được sử dụng để đạt được điều này.

1. **Độ thuần khiết**

Quá trình ta xây cây quyết định nhị phân là quá trình ta phân chia không gian dữ liệu vào (input space). Một cách tiếp cận vét cạn để chia không gian đó chính là chia đôi đệ quy (recursive binary splitting).

Chia đôi đệ quy là thuật toán heuristics, từng đặc trưng một được lựa chọn, từng giá trị ngưỡng được thử với đặc trưng. Các lựa chọn này được đánh giá thông qua hàm mất mát, và thông số tối ưu hàm mất mát được lựa chọn.

Với bài toán phân loại, ta sử dụng độ thuần khiết Gini. Nhắc lại, độ thuần khiết Gini được tính bằng cách lấy tổng xác suất p i của bản ghi i được chọn nhân với xác suất 1 – Pi tức xác suất khi phân loại sai bản ghi i . Độ thuần khiết đạt giá trị nhỏ nhất, tức bằng 0 khi tất cả các trường hợp tại điểm nút cuối đều rơi vào cùng một giá trị.

Độ thuần khiết Gini cũng là độ đo lý thuyết thông tin và tương ứng với entropy dạng Tsallis với hệ số biến dạng q = 2 , trong Vật lý, công thức entropy này đại diện cho tính mất thông tin trong các hệ thống lượng tử mất cân bằng, tắt dần.

Khi có q → 1 , công thức trở về dạng công thức entropy của Shannon. Với các điều kiện này, độ thuần khiết Gini có tác dụng như một biến thể của phép đo entropy cho các cây quyết định.

Entropy là một độ đo độ hỗn loạn hoặc không chắc chắn trong tập dữ liệu.

Đối với bài toán phân loại, entropy được tính bằng công thức:

Entropy =

Trong đó n là số lượng lớp, *pi*​ là tỷ lệ số mẫu thuộc lớp *i* trong tập dữ liệu.

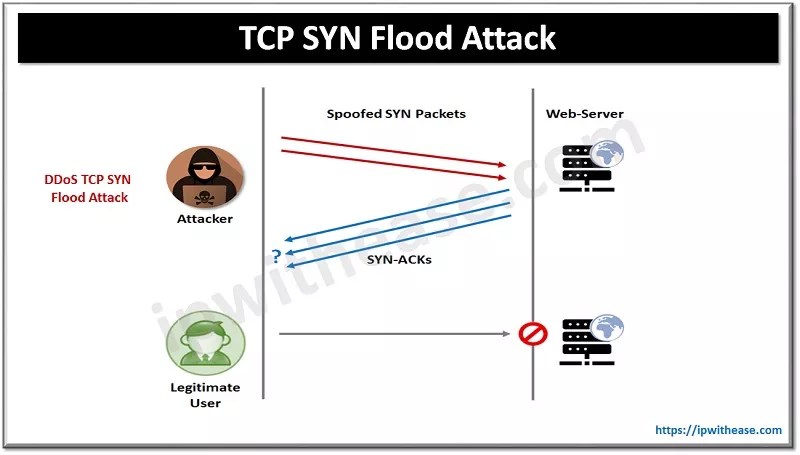
Entropy càng giảm khi dữ liệu trở nên đặc trưng và tinh khiết hơn.

1. **Thực hiện hai dạng tấn công**
2. **Tấn công SYN Flood**
   1. **SYN Flood attack là gì**

Tấn công SYN Flood là một loại tấn công mạng thường được thực hiện để làm quá tải một hệ thống, server, hoặc mạng bằng cách làm cho nó không thể xử lý được lượng yêu cầu kết nối mới. Đây là một dạng tấn công Denial of Service (DoS) hoặc Distributed Denial of Service (DDoS).

Trong tấn công SYN Flood, kẻ tấn công gửi một lượng lớn gói tin SYN (Synchronize) đến một máy chủ mục tiêu. Gói tin SYN được sử dụng trong quá trình thiết lập một kết nối TCP. Khi máy chủ nhận được một yêu cầu kết nối TCP (gói tin SYN), nó sẽ gửi một gói tin ACK (Acknowledgment) trả lời để xác nhận yêu cầu và mở một kết nối. Tuy nhiên, trong tấn công SYN Flood, kẻ tấn công không hoàn thành quá trình thiết lập kết nối bằng cách không gửi gói tin ACK hoặc không hoàn thành quá trình thiết lập kết nối một cách nhanh chóng.

Như một kết quả, máy chủ mục tiêu sẽ giữ lại tất cả các yêu cầu kết nối chưa hoàn thành, chiếm giữ tài nguyên hệ thống và tạo ra một tình trạng quá tải. Các biện pháp phòng ngừa chống lại tấn công SYN Flood bao gồm sử dụng các giải pháp bảo mật mạng như Firewalls, Intrusion Prevention Systems (IPS), và cấu hình TCP/IP để giảm thiểu ảnh hưởng của tấn công này.



* 1. **Các bước thực hiện tấn công SYN Flood**

Gửi Yêu Cầu SYN Giả mạo: Kẻ tấn công bắt đầu bằng cách gửi một lượng lớn gói tin SYN đến máy chủ mục tiêu. Gói tin SYN này thường chứa thông tin giả mạo về địa chỉ nguồn, giả lập các yêu cầu kết nối TCP.

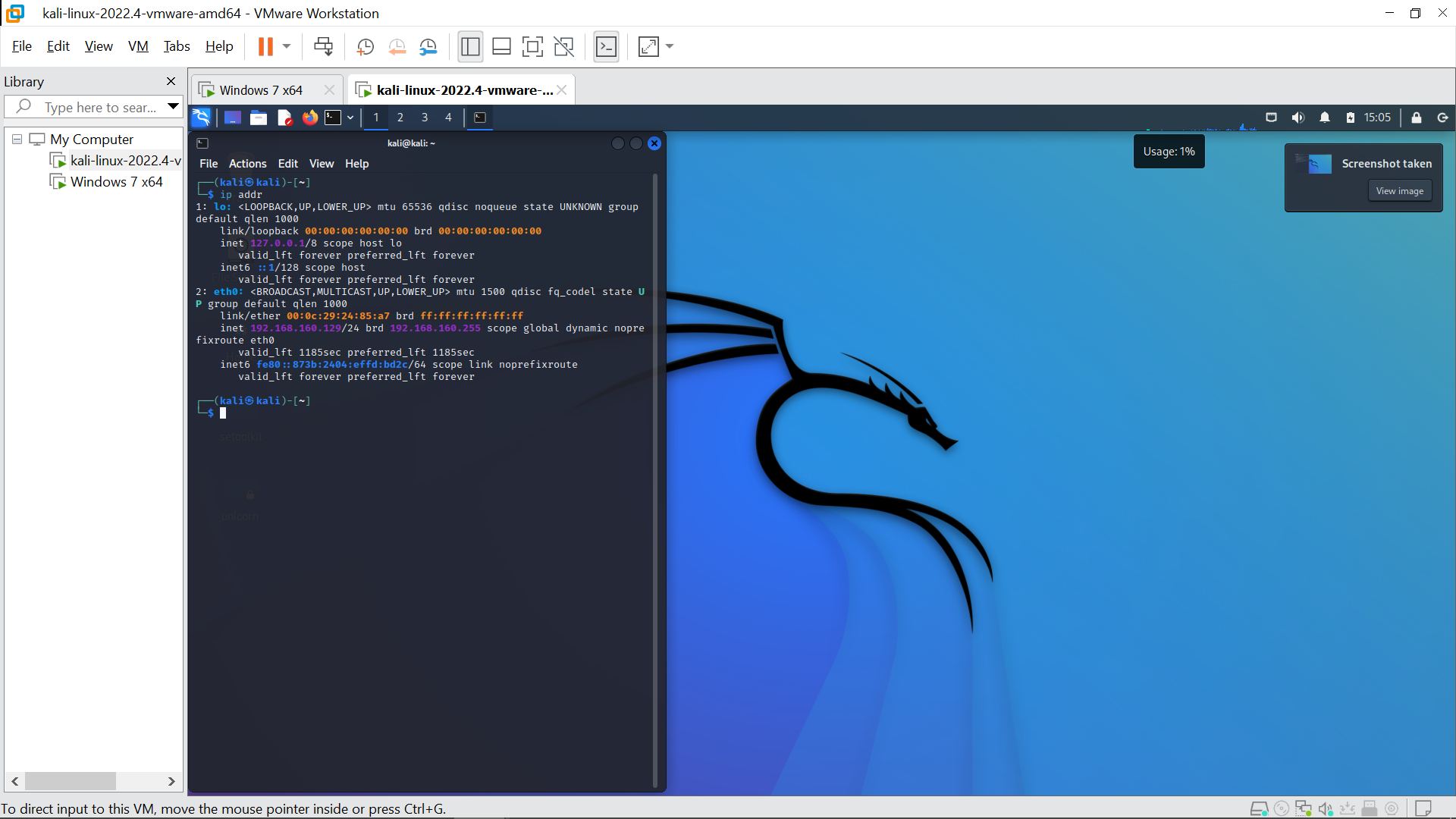
Không Hoàn Thiện Quá Trình Kết Nối: Thay vì hoàn thành quá trình thiết lập kết nối bằng cách gửi gói tin ACK, kẻ tấn công có thể không gửi gói tin ACK hoặc gửi nhanh chóng để không cho máy chủ kịp thời xử lý.

Giữ Nguyên Yêu Cầu Kết Nối Chưa Hoàn Thành: Máy chủ mục tiêu giữ lại các yêu cầu kết nối chưa hoàn thành, do đó chiếm giữ tài nguyên hệ thống và tạo ra tình trạng quá tải.

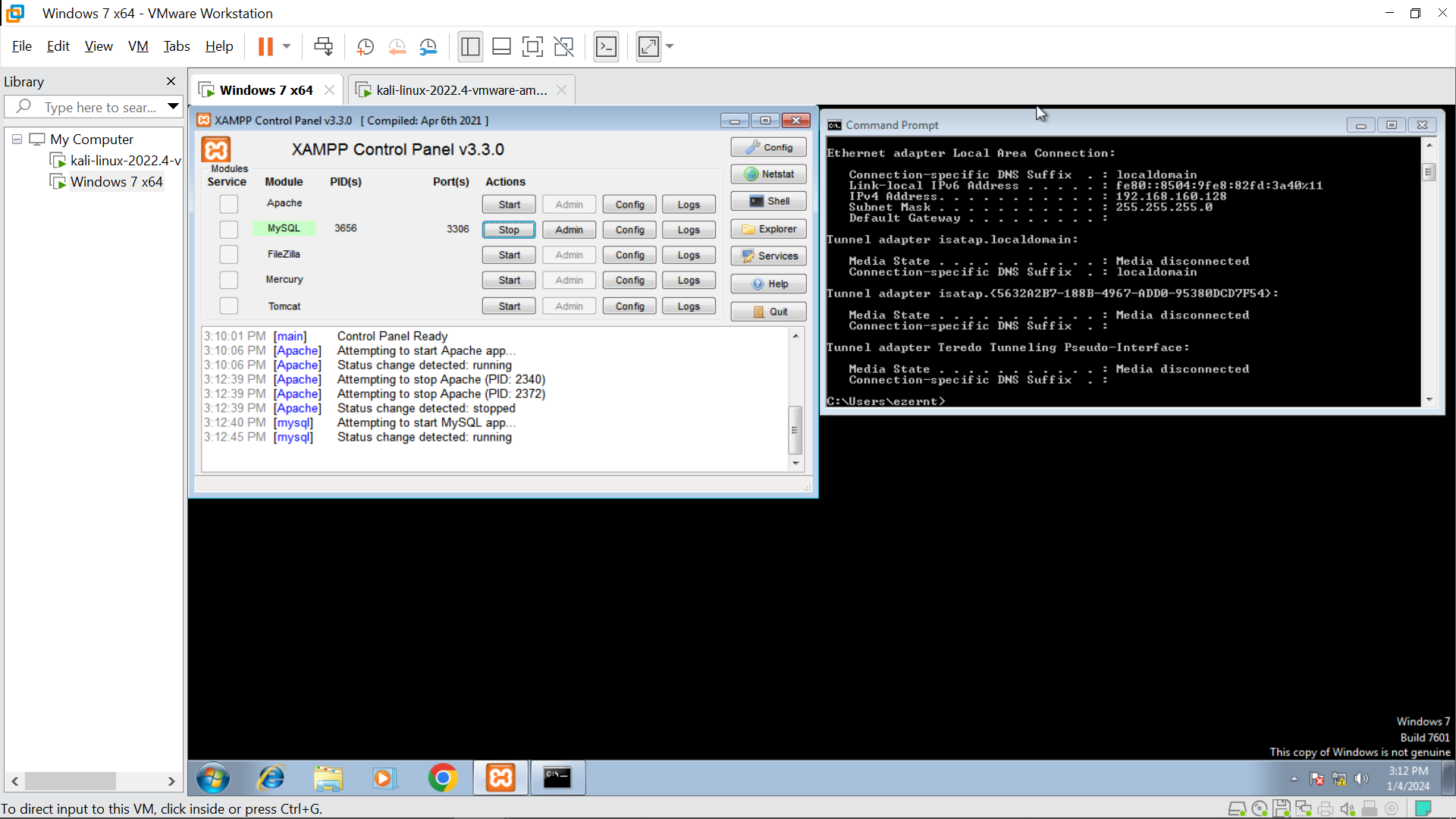
Quá Tải Hệ Thống: Với lượng lớn yêu cầu kết nối chưa hoàn thành, máy chủ mục tiêu trở nên quá tải. Điều này có thể dẫn đến việc từ chối dịch vụ (Denial of Service - DoS) hoặc khi có sự tham gia của nhiều máy chủ, có thể trở thành một cuộc tấn công phân tán (Distributed Denial of Service - DDoS).

Ngăn Chặn Người Dùng Hợp Lệ: Do tất cả các tài nguyên hệ thống được sử dụng để xử lý các yêu cầu kết nối giả mạo, các người dùng hợp lệ có thể trải qua khó khăn khi cố gắng truy cập dịch vụ từ máy chủ bị tấn công.

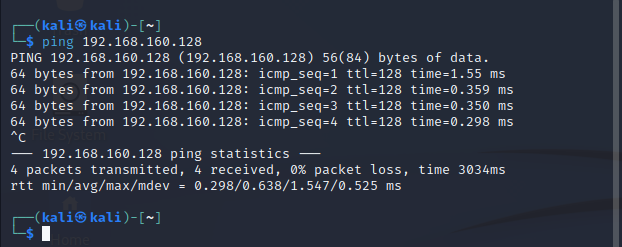
* 1. **Thực hiện**
* Trên ứng dụng máy ảo VMware, triển khai 2 máy ảo gồm:
  + Máy Attacker: là máy ảo sử dụng hệ điều hành kali linux, có địa chỉ ip 192.168.160.129, sử dụng Network Adapter ở trạng thái Host-only.



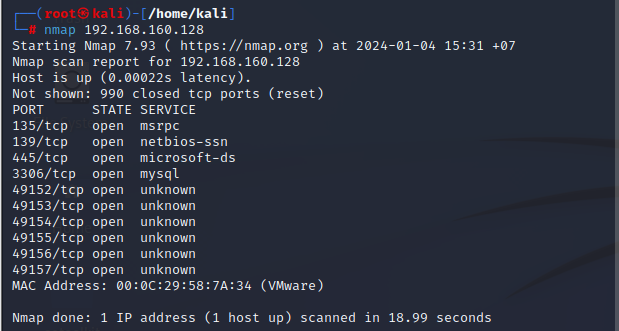
* + Máy Victim: là máy ảo sử dụng hệ điều hành Windows 7 64bit có đía chỉ ip 192.168.160.129, với dịch vụ Mysql đang hoạt động trên port 3306, sử dụng Network Adapter ở trạng thái Host-only.



* Triển khai tấn công trên kali linux:
  + Ping kiểm tra kết nối tới máy victim:



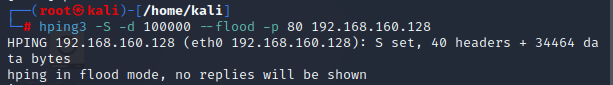
* + Sử dụng nmap để kiểm tra các cổng hoạt động trên máy victim



* + Thực hiện tấn công SYN Flood qua ứng dụng hping3:

Hping3, là một công cụ mạng mạnh mẽ, không chỉ hỗ trợ các chức năng kiểm thử bảo mật cơ bản mà còn cung cấp nhiều tùy chọn tinh vi cho việc tùy chỉnh gói tin và thực hiện các phân tích mạng chi tiết. Điều này giúp người sử dụng có khả năng linh hoạt trong việc điều chỉnh các thử nghiệm và xác định các lỗ hổng bảo mật tiềm ẩn trong hệ thống hay mạng.

Công cụ này có khả năng mô phỏng các kịch bản tấn công phức tạp như tấn công SYN Flood, giúp chuyên gia bảo mật và người quản trị mạng hiểu rõ hơn về khả năng phục hồi của hệ thống trong điều kiện áp lực mạng lớn.



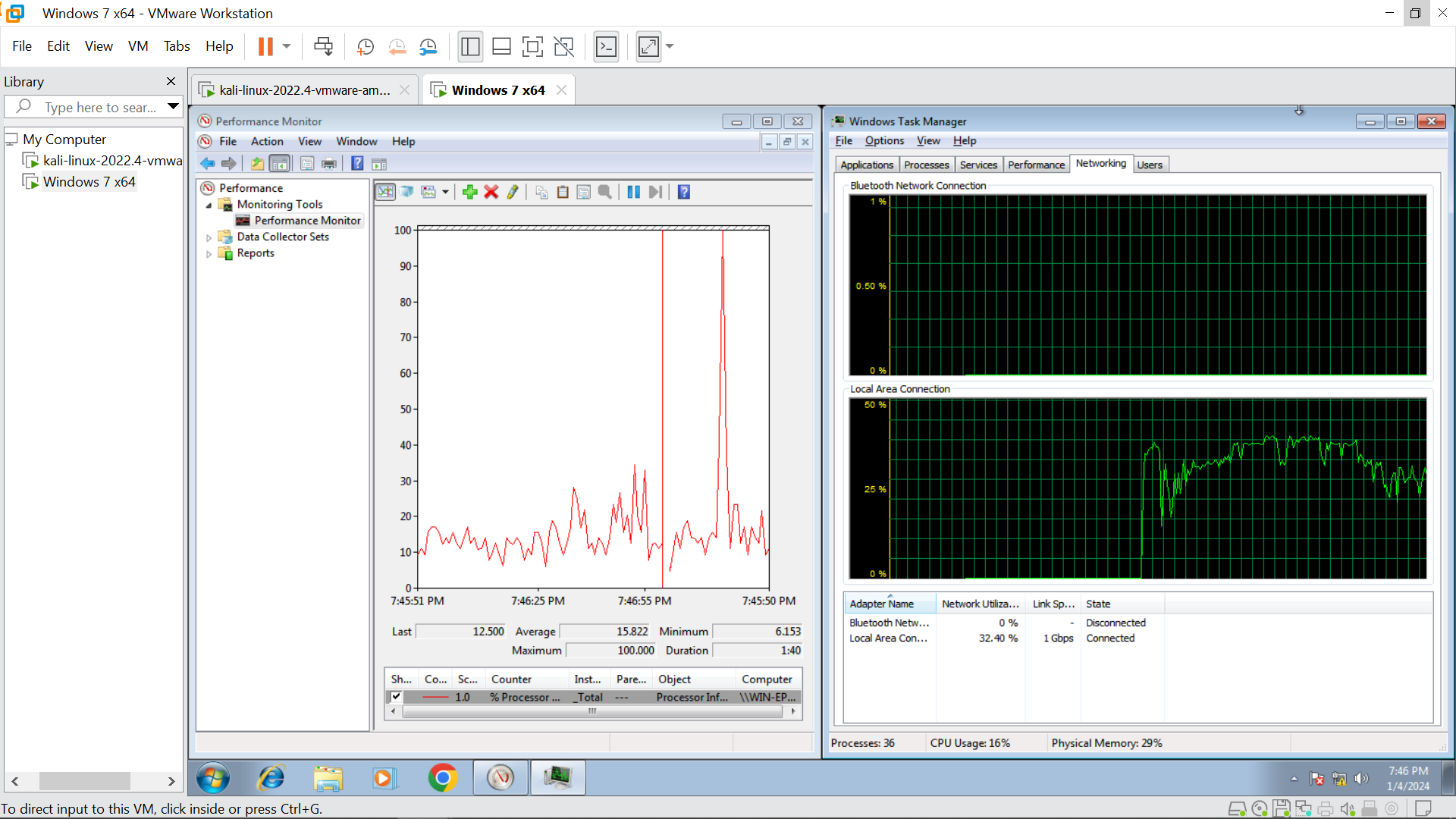
* + Sử dụng ứng dụng Wireshark để xem các gói tin được truyền:

Ta nhận thấy rằng máy attack đang liên tục gửi các gói tin SYN yêu cầu để kết nối TCP với máy victim.

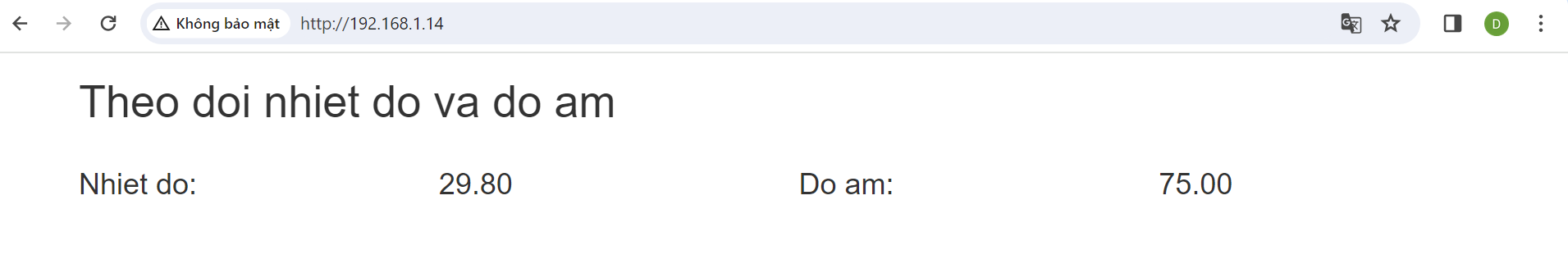
Máy victim gửi lại các gói tin ACK để xác nhận kết nối tuy nhiên không được phản hồi.

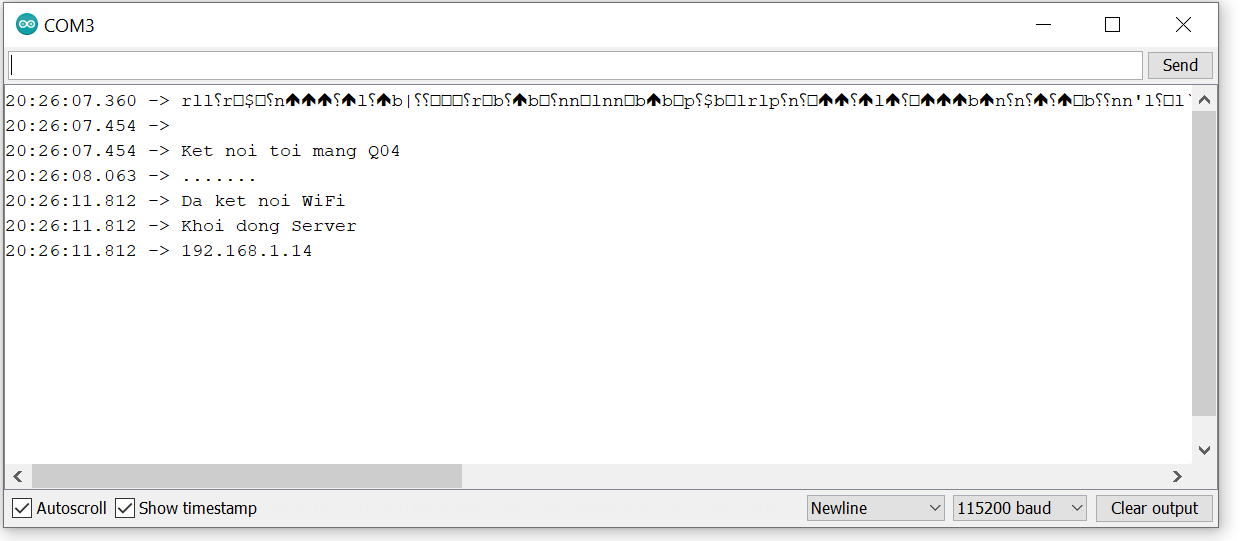


* + Trên máy victim, khi bị tấn công ta thấy được máy victim đang phải sử dụng tài nguyên để đáp ứng lại các gói tin của máy attack.



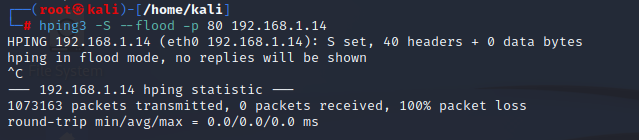
* Cuộc tấn công thứ 2:
  + Triển khai một server web trên thiết bị esp8266 dùng để hiển thị nhiệt độ, độ ẩm khi truy cập với địa chỉ trong mạng LAN là 192.168.1.14



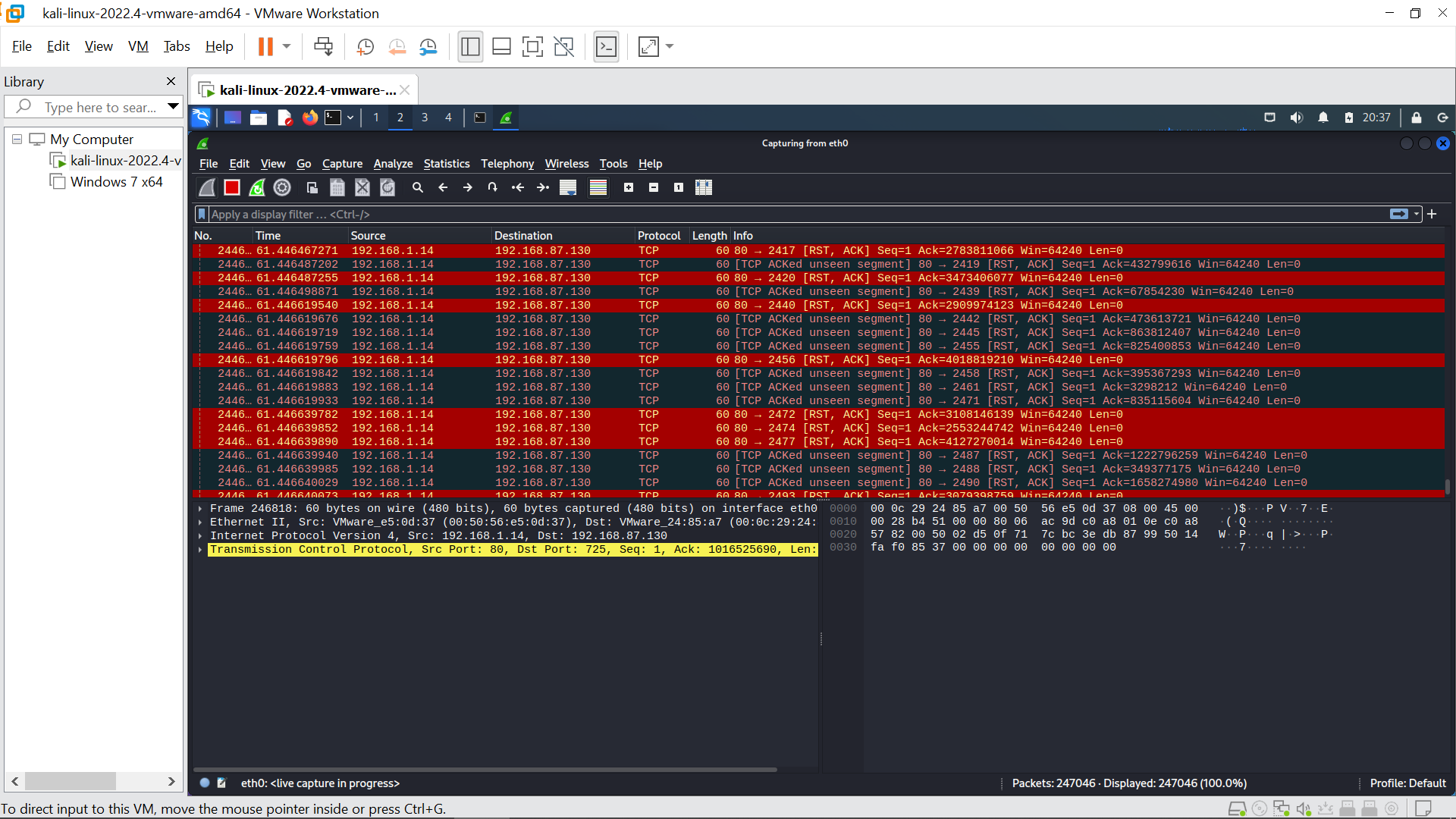


* + Thực hiện tấn công:

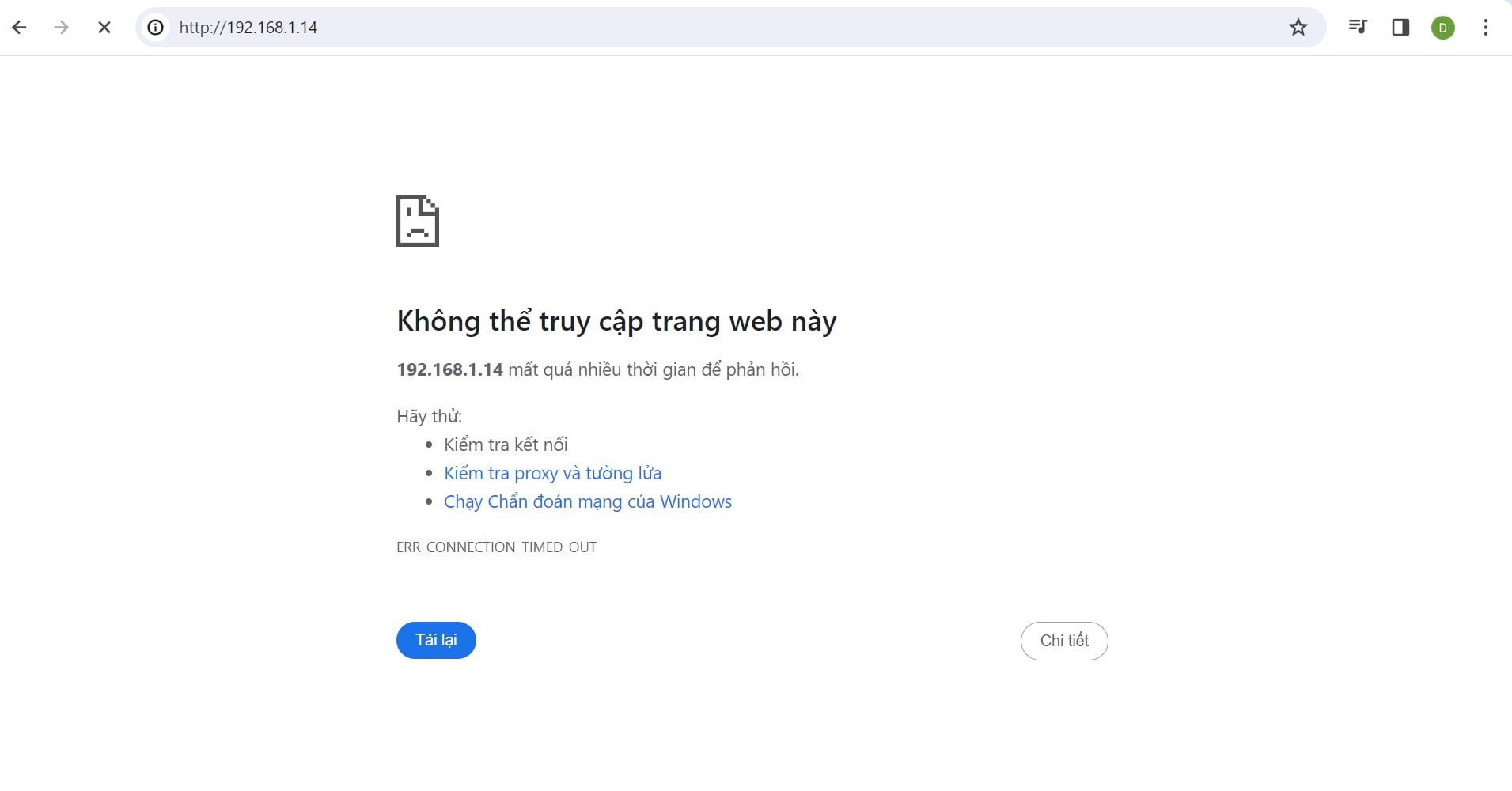
Ip hiên tại của máy attack là 192.168.87.130, với chế độ Network Adapter là NAT



* + Xem các gói tin trên Wireshark:



* + Trạng thái trang web khi đang bị tấn công:

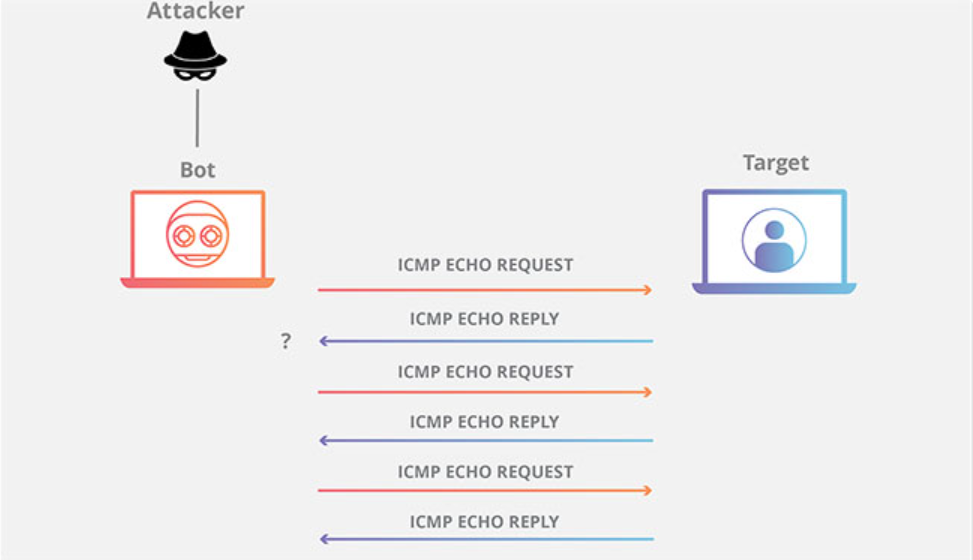


1. **Tấn công ICMP Flood**
   1. **Giới thiệu**

Cuộc tấn công ICMP (Internet Control Message Protocol) Flood, còn được gọi là cuộc tấn công Ping Flood, là một cuộc tấn công từ chối dịch vụ (DoS) phổ biến, trong đó kẻ tấn công cố gắng áp đảo một thiết bị được nhắm mục tiêu bằng ICMP echo-request (ping).

Thông thường, các tin nhắn ICMP echo-request và echo-reply được sử dụng để ping một thiết bị mạng nhằm chẩn đoán tình trạng và kết nối của thiết bị cũng như kết nối giữa người gửi và thiết bị. Bằng cách làm tràn ngập mục tiêu với các gói yêu cầu, mạng buộc phải trả lời với số lượng reply packet (gói phản hồi) bằng nhau. Điều này khiến mục tiêu trở nên không thể truy cập được đối với lưu lượng bình thường.

Các loại tấn công ICMP request khác có thể liên quan đến những công cụ hoặc code tùy chỉnh, chẳng hạn như hping và scapy. Lưu lượng tấn công phát ra từ nhiều thiết bị được coi là tấn công từ chối dịch vụ (DDoS) phân tán. Trong kiểu tấn công DDoS này, cả kênh đến và kênh đi của mạng đều áp đảo, tiêu tốn băng thông đáng kể và dẫn đến hiện tượng dịch vụ từ chối.



*Mô hình tấn công ICMP Flood*

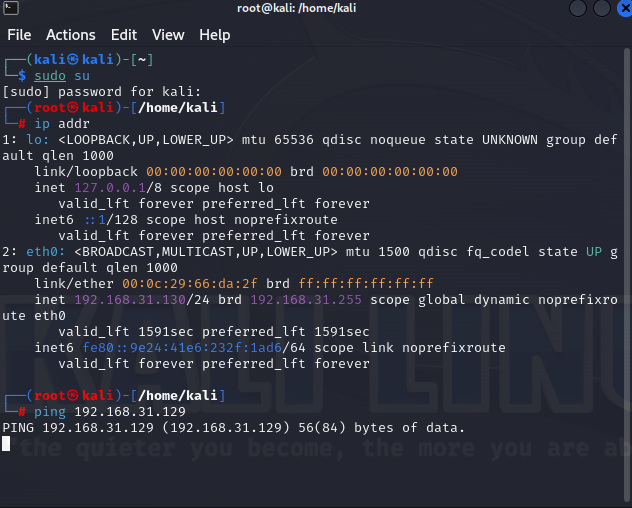
Một cuộc tấn công Ddos ICMP Flood yêu cầu kẻ tấn công phải biết địa chỉ IP của máy nạn nhân.

Ở kĩ thuật này kẻ xấu sẻ gửi một lượng lớn gói tin truy vấn ICMP echo request tới hệ thống máy tính nạn nhân hoặc thông qua một hệ thống trung gian. Tất nhiên theo logic thì máy tính nạn nhân bắt buộc sẻ nhận , xử lý và phản hồi các yêu cầu này thông qua gói tin ICMP echo rely, lúc này máy nạn nhân sẻ căng ra do phải phản hồi liên tục các gói tin echo request, làm ảnh hưởng tới khả năng xử lý

* 1. **Các bước thực hiện**

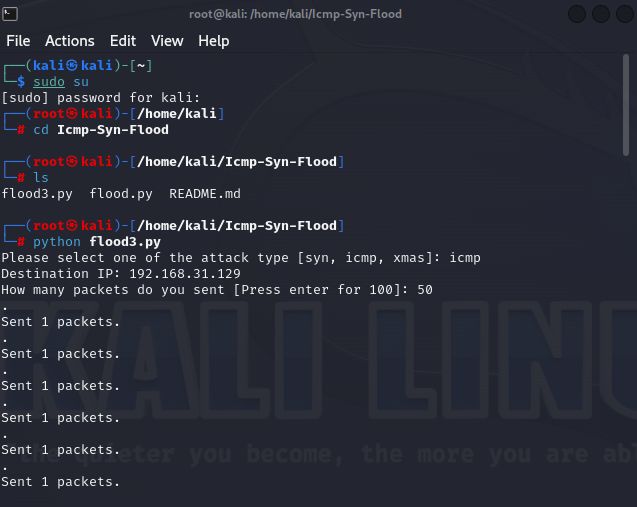
Trên ứng dụng VMWare Tiến hành tạo ra 2 máy ảo:

Máy tấn công : Kali linux có địa chỉ ip 192.168.31.130



-Máy mục tiêu: máy ảo có hệ điều hành window 10 64bit có địa chỉ ip : 192.168.31.129

Tiến hành di chuyển tới file thư mục Icmp-Syn-Flood và chạy file python flood3.py chứa các lệnh tấn công đến địa chỉ ip máy mục tiêu.



File tấn công flood3.py

from scapy.all import \*

def main():

user\_input = input("Please select one of the attack type [syn, icmp, xmas]: ")

if user\_input == "icmp":

icmpflood()

elif user\_input == "syn":

synflood()

elif user\_input == "xmas":

xmasflood()

else:

print("[ERROR] Select one of the attack type !!!")

main()

def icmpflood():

target = destinationIP()

cycle = input("How many packets do you sent [Press enter for 100]: ")

if cycle == "":

cycle = 100

for x in range (0,int(cycle)):

send(IP(dst=target)/ICMP())

def synflood():

target = destinationIP()

targetPort = destinationPort()

cycle = input("How many packets do you sent [Press enter for 100]: ")

if cycle == "":

cycle = 100

for x in range(0, int(cycle)):

send(IP(dst=target)/TCP(dport=targetPort,

flags="S",

seq=RandShort(),

ack=RandShort(),

sport=RandShort()))

def xmasflood():

target = destinationIP()

targetPort = destinationPort()

cycle = input("How many packets do you sent [Press enter for 100]: ")

if cycle == "":

cycle = 100

for x in range(0, int(cycle)):

send(IP(dst=target)/TCP(dport=targetPort,

flags="FSRPAUEC",

seq=RandShort(),

ack=RandShort(),

sport=RandShort()))

def destinationIP():

dstIP = input("Destination IP: ")

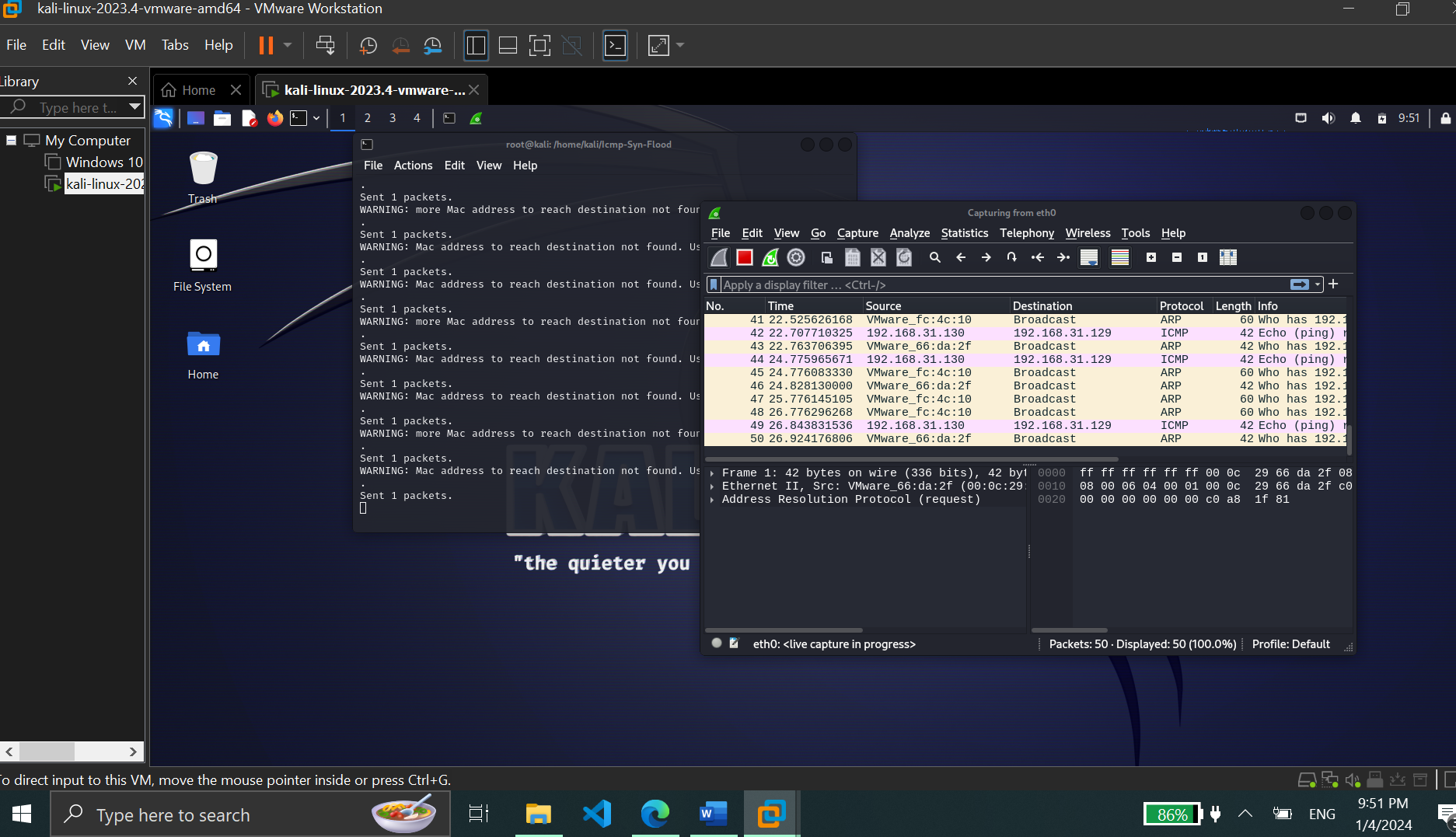
return dstIP

def destinationPort():

dstPort = input("Destination Port: ")

return int(dstPort)

main()



*Xem các gói tin trên WireShark*



*Máy mục tiêu bị tấn công*

1. **Huấn luyện bộ dữ liệu:**
2. **Công nghệ sử dụng:**

*Google Colab*

Google Colab là một phiên bản lưu trữ trên đám mây giống với Jupyter Notebook do Google Research phát triển. Bằng cách sử dụng Google Colab, bạn không cần cài đặt và nâng cấp phần cứng máy tính cá nhân mà vẫn có thể xử lý các công việc nặng về CPU/GPU trong [Python](https://vietnix.vn/python-la-gi/). Colab cung cấp miễn phí quyền truy cập vào các hạ tầng điện toán đám mây như bộ lưu trữ, bộ nhớ, [GPU](https://vietnix.vn/gpu-la-gi/) (đơn vị xử lý đồ hoạ), CPU (khả năng xử lý) và TPU (đơn vị xử lý tensor).

Google đã tạo ra công cụ mã hóa Python dựa trên nền tảng đám mây này để đáp ứng nhu cầu của lập trình viên máy học, nhà khoa học dữ liệu, nhà phân tích dữ liệu, nhà nghiên cứu trí tuệ nhân tạo và người học ngôn ngữ lập trình Python.

* Các thư viện sử dụng trong việc xử lý hình ảnh, trích xuất đặc trưng, tiền xử lý dữ liệu gồm: numpy, pandas, matplotlib.
* Các thư viện sử dụng trong việc huấn luyện dữ liệu: sklearn

*Thư viện Numpy:*

Numpy là một thư viện lõi phục vụ cho khoa học máy tính của Python, hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó.

Numpy đặc biệt hữu ích khi thực hiện các hàm liên quan tới Đại Số Tuyến Tính.

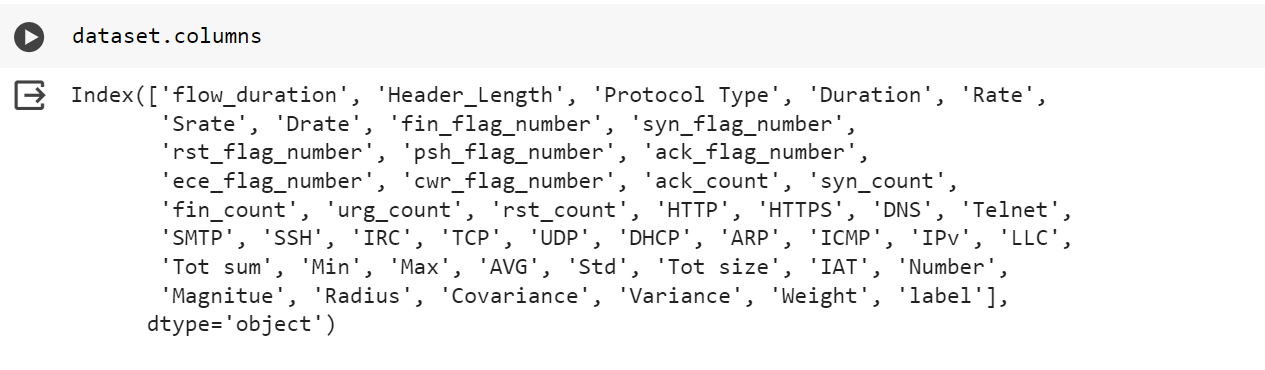
*Thư viện Scikit-learn:*

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.

Trang chủ:<https://scikit-learn.org/>

1. **Thực hiện huấn luyện bằng mô hình học máy Decision Tree**
   1. **Xử lý dữ liệu**

* Dữ liệu đầu vào là các file csv, với tổng cộng 47 cột được liệt kê ở hình dưới.



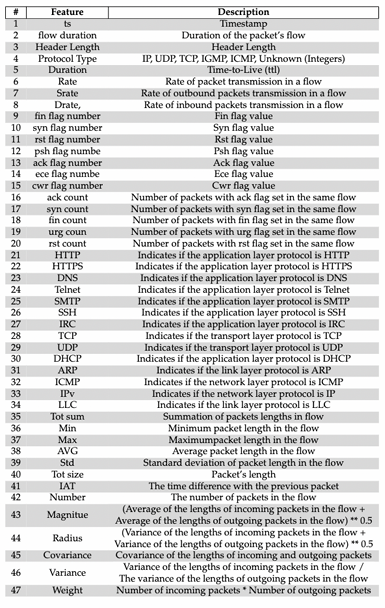
* Tự điển các cột:

Giới thiệu bộ dữ liệu

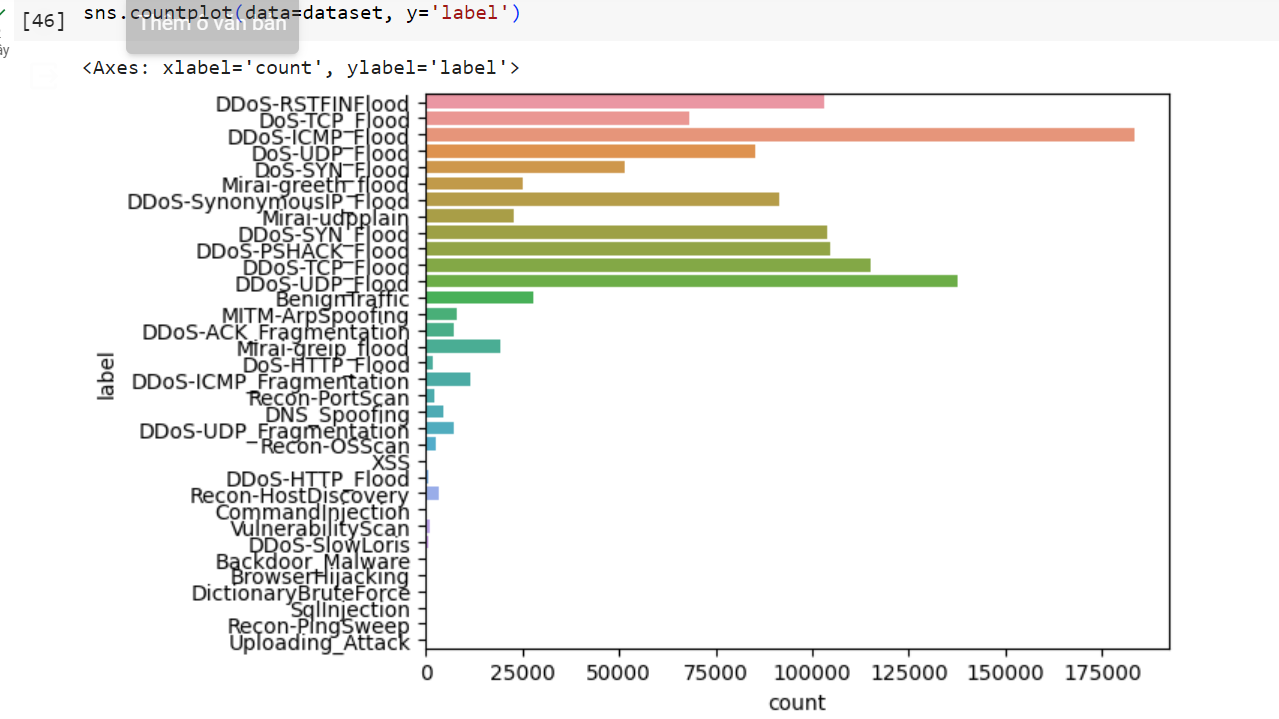
Bộ dữ liệu CICIoT2023 là một bộ dữ liệu thời gian thựcvà chính xác từ các tấn công có quy mô lớn trong môi trường IOT.

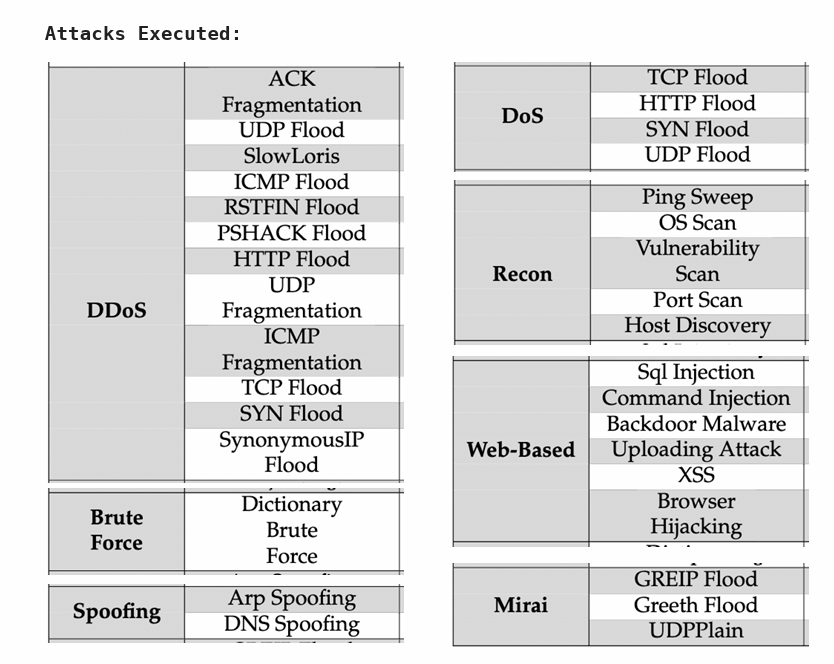
Mục tiêu chính của quá trình nghiên cứu là đưa ra một bộ dữ liệu tấn công Iot góp phần cho quá trình bảo mật an ninh các ứng dụng Iot thực tế,Để hoàn thành nó, đã có tới 33 cuộc tấn công với 105 thiết bị, Các cuộc tấn công được chia làm 7 loại chính: Ddos , Dos, Recon,Web-based, Brute Force, Spoofing and Mirai

Các thuộc tính trong bộ dữ liệu

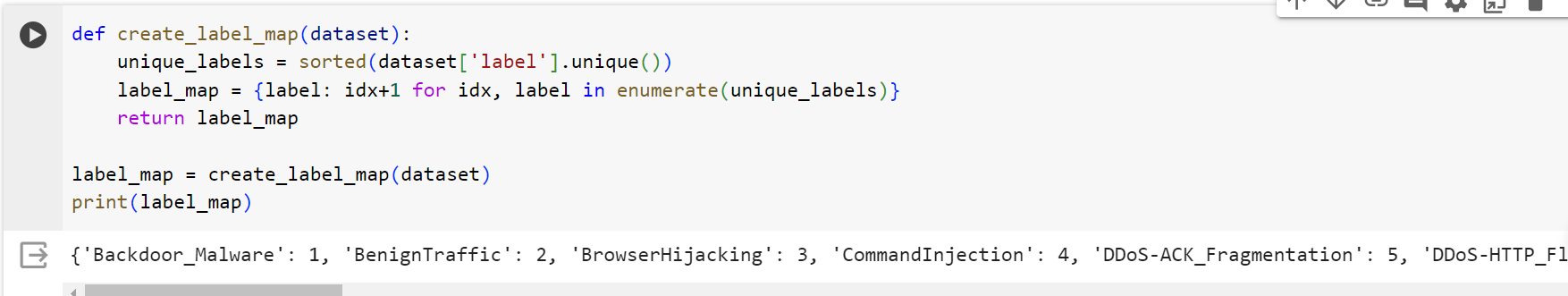


* Các nhãn(label) là các hình thức tấn công trong môi trường IoT với tổng cộng 34 nhãn:

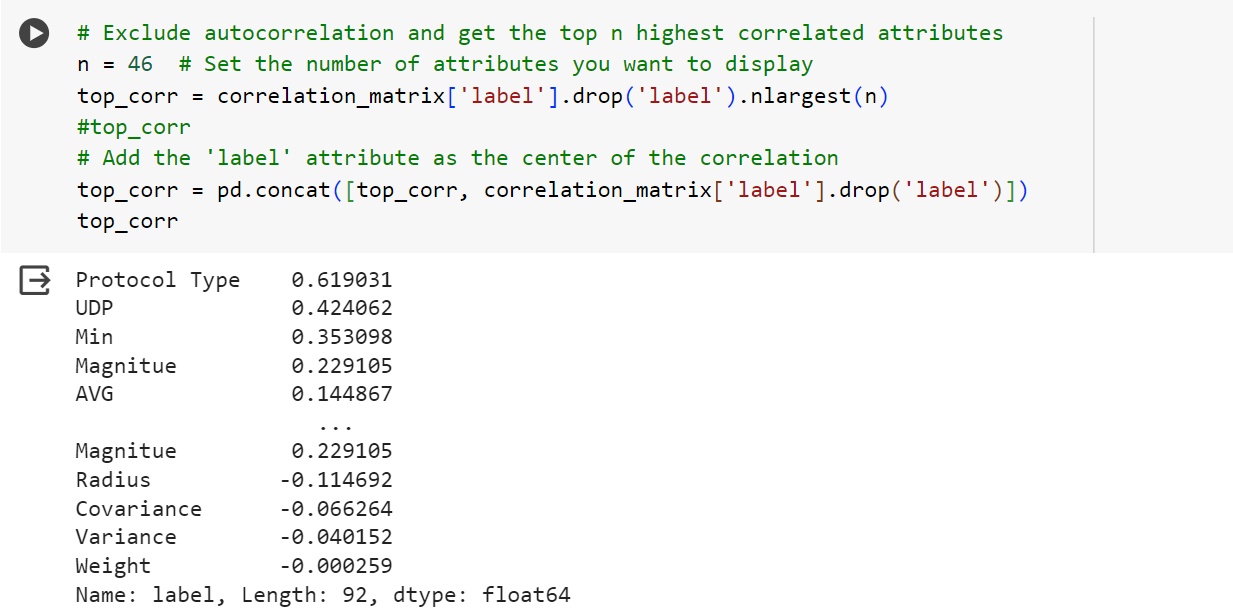




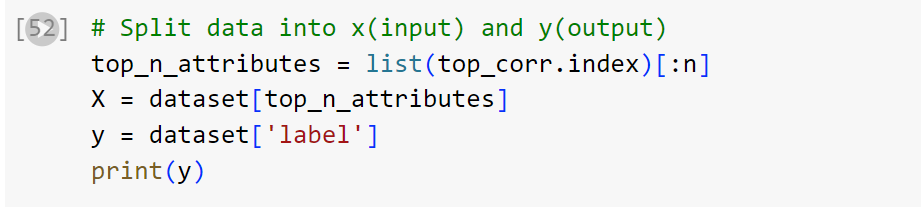
* Đánh số cho các nhãn:



* Xác định, sắp xếp sự tương quan giữa các cột tới nhãn(label):



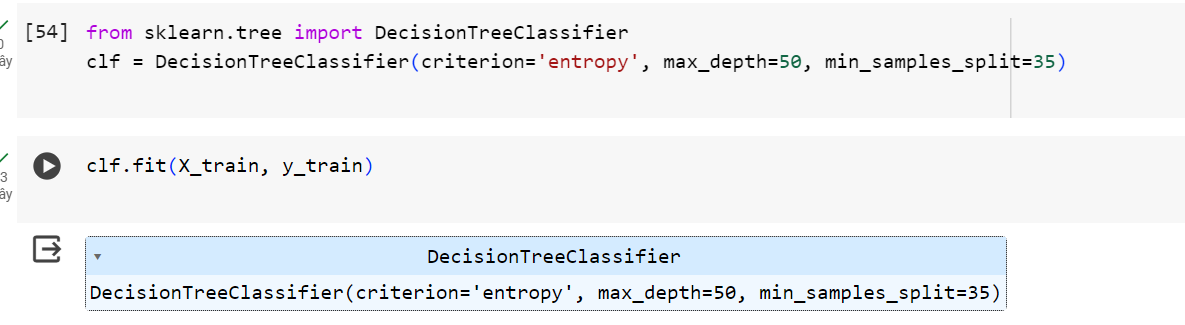
* Tạo tập dữ liệu training
  + X là dataset với các trường được sắp theo độ lớn của sự tương quan với nhãn.
  + y các nhãn.



* 1. Thực hiện training
* Phân chia tập dữ liệu: tập test bằng 5%



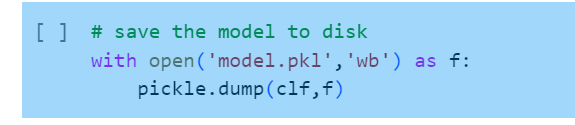
* Sử dụng thư viện sklearn với DecisionTreeClassifier là một triển khai của thuật toán CART được sử dụng cho bài toán phân loại.



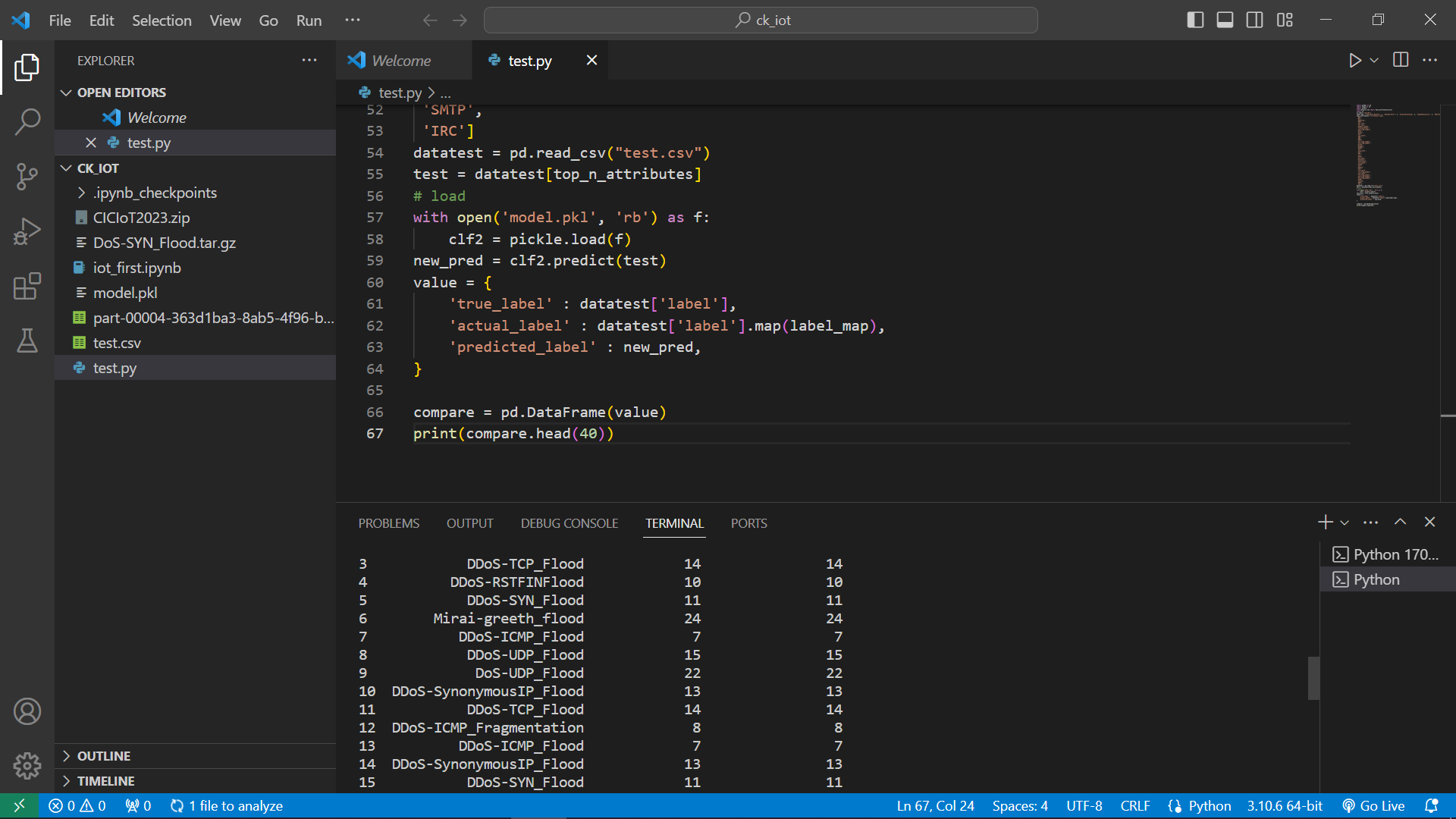
* Đánh giá hiệu suất



* Lưu lại model:



* Sử dụng model đã được lưu:



1. **Kết luận**

Trong bài báo cáo này, chúng tôi đã thực hiện tìm hiểu và phân tích các dạng tấn công mạng trong lĩnh vực IoT. Đối với việc phân loại tấn công, chúng tôi đã triển khai một mô hình Decision Tree để huấn luyện dữ liệu, và kết quả đã cho thấy sự hiệu quả của phương pháp này trong việc phân loại các loại tấn công khác nhau.

Nhờ việc sử dụng một bộ dữ liệu đa dạng, thu thập thông tin từ các thiết bị IoT thực tế để đảm bảo tính thực tế và độ chính xác của mô hình. Quá trình huấn luyện và đánh giá mô hình đã thể hiện khả năng phát hiện tốt của Decision Tree trong việc nhận biết các dạng tấn công, giúp tăng cường an ninh trong môi trường IoT.

Một điểm đáng chú ý là khả năng linh hoạt và diễn giải của Decision Tree, giúp chúng ta hiểu rõ hơn về quy luật và đặc điểm của từng loại tấn công. Điều này có thể hỗ trợ người quản trị mạng và các chuyên gia bảo mật trong việc xác định biện pháp phòng ngừa phù hợp.

Tuy nhiên, chúng tôi nhận thức rằng an ninh IoT là một lĩnh vực đang thay đổi nhanh, vì vậy cần tiếp tục nghiên cứu và cập nhật mô hình với các dữ liệu mới nhất và kỹ thuật tiên tiến.

Bên cạnh đó, chúng tôi đã thành công trong việc thực hiện hai dạng tấn công mạng phổ biến trên môi trường IoT: SYN Flood attack và ICMP attack. Bằng cách mô phỏng các kịch bản tấn công này, chúng tôi đã có cơ hội hiểu rõ hơn về cách chúng có thể ảnh hưởng đến tính ổn định và bảo mật của hệ thống IoT.

Tấn công SYN Flood đã thể hiện khả năng làm quá tải hệ thống thông qua việc tạo ra một lượng lớn các kết nối không hoàn thành, gây ra sự cản trở đối với việc xử lý các yêu cầu hợp lệ. Ngược lại, tấn công ICMP đã tập trung vào lớp mạng, làm tăng gánh nặng cho các thiết bị và có thể dẫn đến mất mát dữ liệu và chậm trễ trong giao tiếp mạng. Việc triển khai demo này không chỉ giúp chúng tôi thấy rõ các hậu quả của các tấn công này mà còn tăng cường kỹ năng phòng ngừa và phản ứng trước những thách thức về an ninh mạng.

Chúng tôi hiểu rằng bảo vệ môi trường IoT đòi hỏi sự chú ý đặc biệt đối với các phương pháp và công nghệ an ninh, cũng như cần thiết phải duy trì sự cảnh báo và sẵn sàng xử lý khi phát hiện các hoạt động đáng ngờ. Những kinh nghiệm thu được từ việc thực hiện demo sẽ giúp thêm hiểu biết cho bản thân về quản trị hệ thống IoT trong việc áp dụng biện pháp bảo mật hiệu quả và đảm bảo tính ổn định của hệ thống trong môi trường đầy thách thức này.