

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

~~~~~\*~~~~~



**ĐỒ ÁN MÔN HỌC**

**Đề tài: Linux kernel networking: Netfilter, IPTables,  
flows of application data packets via TCP/IP protocol stack**

**Môn học: Hệ thống nhúng mạng không dây**

**Sinh viên thực hiện:**

**Tô Công Quân - 22521190**

**Nguyễn Thành Thạo - 22521371**

**Đào Công Sơn - 22521249**

**Giảng viên hướng dẫn:**

**PGS.TS. Lê Trung Quân**

**TP.Hồ Chí Minh - 2024**

## **LỜI CẢM ƠN**

Lời đầu tiên, cho phép tập thể Nhóm 1 đến từ lớp NT131.P13 xin gửi lời cảm ơn và tri ân sâu sắc đến thầy Lê Trung Quân - giảng viên môn Hệ thống nhúng mạng không dây, vì sự tận tâm giúp đỡ, chỉ bảo và hướng dẫn trong quá trình thực hiện đề tài. Những hướng dẫn của thầy đóng vai trò quan trọng trong việc hoàn thiện đồ án này. Chúng em chân thành cảm ơn thầy vì kiến thức và kinh nghiệm mà thầy đã chia sẻ với chúng em.

Chúng em cũng muốn gửi lời cảm ơn đến tất cả các thành viên trong nhóm đồ án. Sự đóng góp và nỗ lực của mỗi người trong việc tìm kiếm tài liệu, đưa ra ý tưởng và hoàn thiện đề tài .

Cuối cùng, vì thời gian và năng lực có hạn nên không tránh khỏi sai sót trong khi thực hiện đồ án học tập của chúng em. Rất mong sự góp ý và bổ sung của thầy và các bạn để đề tài chúng em trở nên hoàn thiện hơn. Một lần nữa, chân thành cảm ơn thầy và tất cả mọi người đã theo dõi đề tài này.

# MỤC LỤC

|                                                                                                                 |      |
|-----------------------------------------------------------------------------------------------------------------|------|
| LỜI CẢM ƠN .....                                                                                                | II   |
| MỤC LỤC .....                                                                                                   | III  |
| DANH MỤC HÌNH ẢNH .....                                                                                         | IV,V |
| TÓM TẮT .....                                                                                                   | VII  |
| I. TỔNG QUAN .....                                                                                              | 1    |
| 1. Iptables .....                                                                                               | 1    |
| 1.1. Vai trò .....                                                                                              | 1    |
| 1.2. Tính năng .....                                                                                            | 1    |
| 1.3. Các loại Iptables và chains .....                                                                          | 1    |
| 1.4. Target .....                                                                                               | 3    |
| 2. Netfilter .....                                                                                              | 3    |
| 2.1. Khái niệm .....                                                                                            | 3    |
| 2.2. Netfilter hook .....                                                                                       | 3    |
| 3. TCP/IP protocol stack .....                                                                                  | 5    |
| II. MÔ HÌNH .....                                                                                               | 6    |
| III. TRIỂN KHAI .....                                                                                           | 6    |
| 1. Netfilter .....                                                                                              | 6    |
| Kịch bản 1: Hook LOCAL_IN - Giới hạn số lượng kết nối SSH (Chỉ có 1 kết nối bất kỳ) .....                       | 6    |
| Kịch bản 2: Chặn các máy muốn ping hoặc telnet đến một host cụ thể (10.0.3.8) .....                             | 9    |
| Kịch bản 3: Chặn các gói tin từ một địa chỉ IP nhất định khi cố gắng truy cập Apache port 80 trên máy chủ ..... | 15   |
| 2. Iptables .....                                                                                               | 18   |
| Kịch bản 4: OUTPUT - Chặn truy cập Internet .....                                                               | 18   |
| Kịch bản 5: INPUT - Ghi log và gửi cảnh báo về mail khi bị scanning port .....                                  | 20   |
| Kịch bản 6: Từ chối mọi kết nối chỉ giữ lại ping .....                                                          | 26   |
| Kịch bản 7: Chống DOS .....                                                                                     | 28   |
| Kịch bản 8: Chỉ cho phép một địa chỉ cụ thể ssh vào host .....                                                  | 29   |
| Kịch bản 9: Chặn kết nối HTTP .....                                                                             | 33   |
| IV. KẾT LUẬN .....                                                                                              | 35   |
| BẢNG PHÂN CÔNG .....                                                                                            | 37   |
| TAI LIỆU THAM KHẢO .....                                                                                        | 37   |

# DANH MỤC HÌNH ẢNH

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| Sơ đồ minh họa .....                                                      | 4  |
| Kiến trúc Netfilter .....                                                 | 6  |
| Kiến trúc iptables .....                                                  | 6  |
| Kịch bản 1: Hàm giới hạn số lượng ssh .....                               | 7  |
| Kịch bản 1: Hook được gắn ở LOCAL_IN .....                                | 7  |
| Kịch bản 1: SSH tới host bằng Windows .....                               | 8  |
| Kịch bản 1: SSH tới host bằng Ubuntu .....                                | 8  |
| Kịch bản 1: Builld .....                                                  | 9  |
| Kịch bản 1: SSH từ máy Windows và đọc log .....                           | 9  |
| Kịch bản 1: Log của máy Ubuntu ssh tới host .....                         | 9  |
| Kịch bản 2: Hàm block ICMP .....                                          | 10 |
| Kịch bản 2: Hàm block Telnet .....                                        | 11 |
| Kịch bản 2: Các hook được đăng ký .....                                   | 11 |
| Kịch bản 2: Hàm printInfo .....                                           | 12 |
| Kịch bản 2: Hàm registerFilter và hàm removeFilter .....                  | 13 |
| Kịch bản 2: Ping và telnet đến 10.0.3.8 trước khi áp dụng Netfilter ..... | 14 |
| Kịch bản 2: Build và cài đặt Netfilter module vào kernel .....            | 14 |
| Kịch bản 2: Thực hiện ping đến 10.0.3.8 sau khi áp dụng Netfilter .....   | 15 |
| Kịch bản 2: Thực hiện telnet đến 10.0.3.8 sau khi áp dụng Netfilter ..... | 15 |
| Kịch bản 3: Thiết lập địa chỉ ip và port mục tiêu .....                   | 16 |
| Kịch bản 3: Hàm chặn gói tin .....                                        | 16 |
| Kịch bản 3: Hàm khởi tạo và hủy bỏ module .....                           | 17 |
| Kịch bản 3: Gửi yêu cầu HTTP đến host .....                               | 17 |
| Kịch bản 3: Biên dịch và nạp module .....                                 | 18 |
| Kịch bản 3: Gửi yêu cầu sau khi nạp module .....                          | 18 |
| Kịch bản 3: Xem log .....                                                 | 18 |
| Kịch bản 4: Kiểm tra các rules của iptables .....                         | 19 |
| Kịch bản 4: Kiểm tra truy cập Internet trước khi có rules .....           | 19 |
| Kịch bản 4: Áp dụng rules .....                                           | 20 |
| Kịch bản 4: Kiểm tra truy cập sau khi áp dụng rules .....                 | 20 |
| Kịch bản 5: Thêm rules iptables .....                                     | 21 |
| Kịch bản 5: Chính sửa email nhận .....                                    | 21 |
| Kịch bản 5: Bị chặn port 25 .....                                         | 21 |
| Kịch bản 5: Thiết lập cho phép port 587 .....                             | 22 |
| Kịch bản 5: Thiết lập email và mật khẩu .....                             | 23 |
| Kịch bản 5: Cấp quyền bảo mật và khởi động lại .....                      | 23 |
| Kịch bản 5: Thay đổi tiêu đề mail .....                                   | 23 |

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| Kịch bản 5: Quét cổng host sử dụng nmap .....                            | 24 |
| Kịch bản 5: Gửi email ngay lập tức .....                                 | 24 |
| Kịch bản 5: Xem email .....                                              | 25 |
| Kịch bản 5: Xem log của mail .....                                       | 25 |
| Kịch bản 5: Email được gửi sau một ngày .....                            | 26 |
| Kịch bản 6: Thêm các quy tắc .....                                       | 26 |
| Kịch bản 6: Kết quả ping khi các quy tắc được áp dụng .....              | 27 |
| Kịch bản 6: Sử dụng netcat để kết nối khi các quy tắc được áp dụng ..... | 28 |
| Kịch bản 6: Kết quả là không thể kết nối được .....                      | 28 |
| Kịch bản 7: Thiết lập quy tắc .....                                      | 29 |
| Kịch bản 8: Địa chỉ ip máy thật windows .....                            | 29 |
| Kịch bản 8: Địa chỉ ip máy ảo windows .....                              | 30 |
| Kịch bản 8: Địa chỉ ip host .....                                        | 30 |
| Kịch bản 8: SSH từ máy ảo windows đến host .....                         | 31 |
| Kịch bản 8: SSH từ máy thật windows đến host .....                       | 31 |
| Kịch bản 8: Kiểm tra các quy tắc .....                                   | 32 |
| Kịch bản 8: Thiết lập quy tắc chặn ip .....                              | 32 |
| Kịch bản 8: SSH từ máy ảo windows đến host lần nữa .....                 | 32 |
| Kịch bản: SSH từ máy thật windows đến host .....                         | 33 |
| Kịch bản 9: Kiểm tra địa chỉ ip máy host .....                           | 33 |
| Kịch bản 9: Kiểm tra địa chỉ ip máy client .....                         | 34 |
| Kịch bản 9: Truy cập website Apache trên client .....                    | 34 |
| Kịch bản 9: Thiết lập quy tắc .....                                      | 35 |
| Kịch bản 9: Kết quả sau khi chặn .....                                   | 35 |

## TÓM TẮT

Netfilter là một framework mạnh mẽ trong Linux kernel, cho phép lọc, theo dõi và thao tác các gói tin mạng khi chúng đi qua hệ thống. Nó hoạt động tại các điểm hook trong stack TCP/IP, cung cấp cơ chế để thay đổi hoặc loại bỏ gói tin dựa trên các quy tắc xác định. IPTables là công cụ giao diện người dùng để tương tác với Netfilter, giúp quản trị viên thiết lập các quy tắc tường lửa, NAT, và chuyển tiếp gói tin, kiểm soát luồng dữ liệu vào, ra và qua hệ thống. Gói tin ứng dụng được truyền tải thông qua các tầng của giao thức TCP/IP, trong đó tầng mạng đảm bảo định tuyến, tầng transport (như TCP hoặc UDP) đảm bảo kết nối và độ tin cậy, trước khi dữ liệu được đóng gói và truyền qua các interface mạng. Các cơ chế này giúp duy trì an ninh và hiệu suất mạng trong các hệ thống Linux.

# I. TỔNG QUAN

## 1. Iptables

### 1.1. Vai trò

Iptables là một ứng dụng dùng để quản lý filtering gói tin và NAT rules hoạt động trên console của linux rất nhỏ và tiện dụng. Được cung cấp miễn phí nhằm nâng cao tính bảo mật trên hệ thống Linux.

Iptables bao gồm hai phần là netfilter nằm bên trong nhân Linux và iptables nằm ở vùng ngoài nhân. Iptables chịu trách nhiệm giao tiếp với người dùng và sau đó đẩy rules của người dùng vào cho netfilter xử lý. Netfilter thực hiện công việc lọc các gói tin ở mức IP. Netfilter làm việc trực tiếp ở trong nhân của Linux nhanh và không làm giảm tốc độ của hệ thống.

### 1.2. Tính năng

- Có khả năng phân tích gói tin hiệu quả.
- Filtering gói tin dựa vào MAC và một số cờ hiệu (flags) trong TCP Header.
- Cung cấp kỹ thuật NAT, chi tiết cho các tùy chọn để ghi nhận sự kiện hệ thống.
- Có khả năng ngăn chặn một số cơ chế tấn công theo kiểu DoS.
- Xây dựng một hệ thống tường lửa (firewall).
- Cung cấp, xây dựng và quản lý các rule để xử lý các gói tin.

### 1.3. Các loại Iptables và chains

#### 1.3.1. Iptables

- NAT table: Cho phép route các gói tin đến các host khác nhau trong mạng bằng cách thay đổi IP nguồn và IP đích của gói tin. Table này quy định và cho phép các kết nối có thể truy cập tới các dịch vụ không được truy cập trực tiếp.
- Filter table: được sử dụng mặc định bởi iptables khi bạn tạo các chain mà không khai báo cho chain đó thuộc vào table nào. Table hoạt động với việc quy định việc quyết định có cho phép gói tin được chuyển đến địa chỉ đích hay không
- Mangle table: Table này liên quan đến việc sửa header của gói tin, ví dụ chỉnh sửa giá trị các trường TTL, MTU, Type of Service.

- Raw table: Bảng này được sử dụng chủ yếu dành cho việc cấu hình sử dụng chain có sẵn
- Security table: Đây là bảng được sử dụng cho Mandatory Access Control (MAC) - kiểm soát truy cập bắt buộc đối với các rule về network.

### 1.3.2. Chain

**Chain** là một quy tắc xử lý các gói tin bao gồm nhiều rules có liên quan tới nhau.

Trong IPTables, các **rule** sẽ được sắp xếp gán vào các **table** (bảng). Các table này mục đích là để phân loại **rule** theo mục đích. Ví dụ - Các rule được viết với mục đích allow hoặc block gói tin, thường sẽ được xếp vào table filter. Ngoài ra có các table khác như: NAT, mangle, security,... Trong mỗi table, các rule sẽ được tiếp tục được chia ra thành nhiều **chain** riêng. Các chain này sẽ thể hiện netfilter hook nào sẽ kích hoạt rule. Do đó, tên các chain cũng khá tương đồng với tên các netfilter hook:

**PREROUTING**: Tương ứng với hook **NF\_IP\_PRE\_ROUTING** - Các rule thuộc chain này sẽ được áp dụng trong quá trình xử lý gói tin trước khi nó được định tuyến. **Chain** này chỉ có thể có ở table NAT, RAW và MANGLE.

**INPUT**: Tương ứng với hook **NF\_IP\_LOCAL\_IN** - Các rule thuộc chain này sẽ áp dụng cho các gói tin vào hệ thống từ mạng bên ngoài. Chain này có trong table MANGLE và FILTER.

**FORWARD**: Tương ứng với hook **NF\_IP\_FORWARD** - Các rule thuộc chain này áp dụng các gói tin được chuyển tiếp qua hệ thống. Chain có trong table MANGLE.

**OUTPUT**: Tương ứng với hook **NF\_IP\_LOCAL\_OUT** - Các rule thuộc chain này áp dụng ngay cho các gói tin đi ra từ hệ thống. Chain có trong table MANGLE, RAW và FILTER.

**POSTROUTING**: Tương ứng với hook **NF\_IP\_POST\_ROUTING** - Các rule thuộc chain này áp dụng trong quá trình xử lý gói tin sau khi nó đã được định tuyến. Chain này có trong table MANGLE và NAT.

Nhờ sử dụng các **chain** này, sysadmin có thể kiểm soát được khi nào rule sẽ được áp dụng lên gói tin. Như vậy IPTables là giao diện người dùng, cho phép ta có thể “đăng ký” vào các netfilter hook chạy ở kernel, để tương tác với các gói tin đi qua hệ thống.

## 1.4. Target

Mỗi một chain là một danh sách các luật có thể được thiết lập cho các gói tin. Mỗi một luật sẽ cần phải khai báo những gì cần phải làm với gói tin được gọi là **target**.

Nói một cách đơn giản thì các hành động áp dụng cho các gói tin được gọi là **target**. Đối với những gói tin đúng theo rule mà chúng ta đặt ra thì các hành động (target) có thể thực hiện được đó là:

**ACCEPT**: chấp nhận gói tin, cho phép gói tin đi qua hay đi vào hệ thống.

**DROP**: loại bỏ gói tin, không phản hồi lại gói tin giống như việc gói tin đó được gửi đến một hệ thống không tồn tại.

**RETURN**: Dừng thực thi xử lý rule tiếp theo trong chain hiện tại đối với gói tin. Việc kiểm soát sẽ được trả về đối với chain đang gọi.

**REJECT**: Thực hiện loại bỏ gói tin và gửi lại gói tin phản hồi thông báo lỗi. Ví dụ: 1 bản tin “connection reset” đối với gói TCP hoặc bản tin “destination host unreachable” đối với gói UDP và ICMP.

**LOG**: Chấp nhận gói tin và có ghi lại log.

## 2. Netfilter

### 2.1. Khái niệm

Netfilter là một framework được cung cấp bởi nhân Linux, cho phép thực hiện các hoạt động liên quan đến mạng dưới dạng các bộ xử lý tùy chỉnh. Netfilter cung cấp nhiều chức năng và hoạt động cho việc lọc gói tin, chuyển đổi địa chỉ mạng (NAT) và chuyển đổi cổng, cung cấp các tính năng cần thiết để định hướng gói tin qua mạng và ngăn chặn các gói tin tiếp cận những vị trí nhạy cảm trong mạng.

Netfilter đại diện cho một tập hợp các hook bên trong nhân Linux, cho phép các mô-đun nhân (kernel) cụ thể đăng ký các hàm callback với ngăn xếp mạng của nhân (kernel). Các hàm này, thường được áp dụng cho lưu lượng dưới dạng các quy tắc lọc và sửa đổi, sẽ được gọi cho mọi gói tin đi qua hook tương ứng trong ngăn xếp mạng.

### 2.2. Netfilter hook

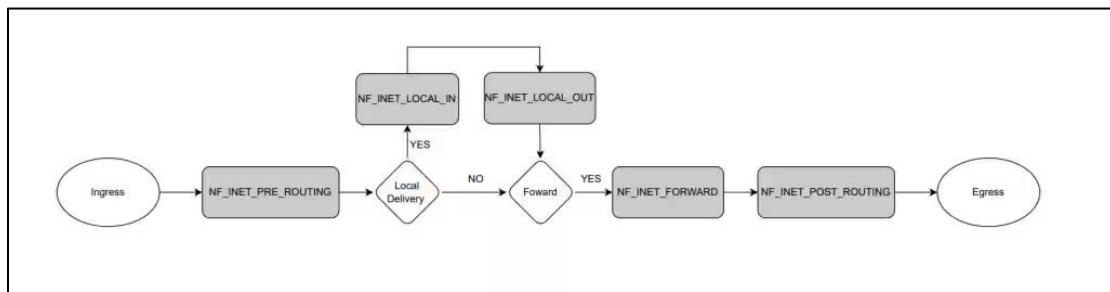
Có năm hook của netfilter mà các chương trình có thể đăng ký. Khi các gói tin tiến qua ngăn xếp mạng, chúng sẽ kích hoạt các mô-đun nhân Linux đã được đăng ký với các hook này. Những hook mà gói tin sẽ kích hoạt phụ thuộc vào

việc gói tin là đến hay đi, đích đến của gói tin, và liệu gói tin có bị loại bỏ hoặc từ chối ở điểm trước đó hay không.

Các hook sau đây đại diện cho những điểm xác định rõ ràng trong ngăn xếp mạng:

- **NF\_IP\_PRE\_ROUTING**: Hook này sẽ được kích hoạt bởi bất kỳ lưu lượng nào đến rất sớm sau khi đi vào ngăn xếp mạng. Hook này được xử lý trước khi có bất kỳ quyết định định tuyến nào về nơi sẽ gửi gói tin.
- **NF\_IP\_LOCAL\_IN**: Hook này được kích hoạt sau khi một gói tin đến đã được định tuyến nếu gói tin đó có đích đến là hệ thống cục bộ.
- **NF\_IP\_FORWARD**: Hook này được kích hoạt sau khi một gói tin đến đã được định tuyến nếu gói tin đó sẽ được chuyển tiếp đến một máy khác.
- **NF\_IP\_LOCAL\_OUT**: Hook này được kích hoạt bởi bất kỳ lưu lượng đi nào được tạo ra từ hệ thống cục bộ ngay khi nó chạm vào ngăn xếp mạng.
- **NF\_IP\_POST\_ROUTING**: Hook này được kích hoạt bởi bất kỳ lưu lượng đi hoặc chuyển tiếp nào sau khi định tuyến đã diễn ra và ngay trước khi được gửi ra ngoài mạng.

Mỗi điểm hook tương ứng với một giai đoạn khác nhau trong quá trình xử lý gói tin, như được minh họa trong sơ đồ dưới đây:



Khi một gói tin đến hoặc rời khỏi giao diện mạng, nó sẽ đi qua từng điểm hook này theo thứ tự. Tại mỗi điểm hook, nhân hệ điều hành gọi tất cả các hàm hook netfilter đã đăng ký tại điểm đó. Mỗi hàm hook netfilter có thể kiểm tra gói tin và quyết định xử lý nó như thế nào. Các hành động có thể là:

- **Accept (Chấp nhận)**: Gói tin được phép tiếp tục đến điểm hook tiếp theo hoặc đích đến của nó.
- **Drop (Loại bỏ)**: Gói tin bị loại bỏ một cách âm thầm và không có xử lý thêm nào được thực hiện.

- **Queue (Xếp hàng):** Gói tin được đưa vào hàng đợi để xử lý bởi không gian người dùng thông qua các trình xử lý như iptables hoặc nftables.
- **Repeat (Lặp lại):** Gói tin được tái tiêm vào tại cùng điểm hook để xử lý thêm một vòng nữa.
- **Stop (Dừng):** Gói tin được chấp nhận và không có xử lý thêm nào khác được thực hiện.

Điều này cho phép linh hoạt trong việc xử lý và lọc gói tin tại mỗi giai đoạn trong chuỗi xử lý mạng.

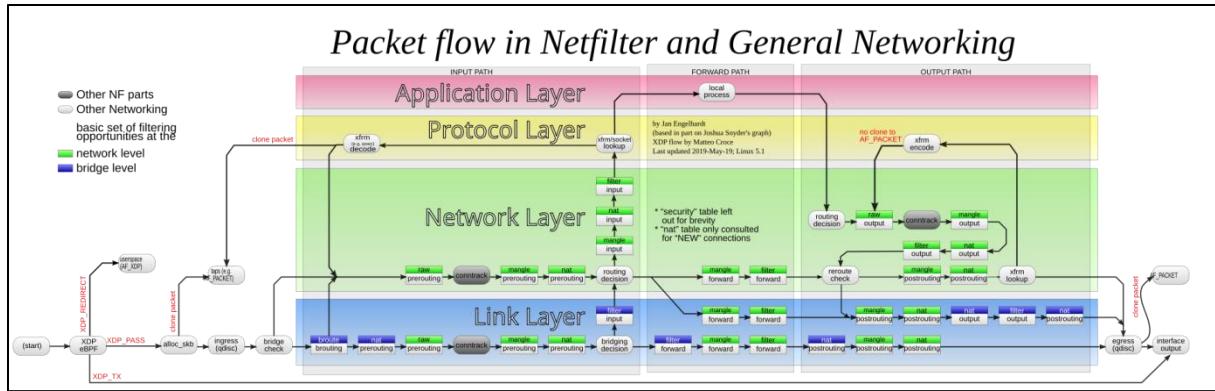
**Lưu ý:** Các mô-đun nhân Linux cần đăng ký với các hook này cũng phải cung cấp một số thứ tự tiên để giúp xác định thứ tự chúng sẽ được gọi khi hook được kích hoạt. Điều này cho phép nhiều mô-đun (hoặc nhiều phiên bản của cùng một mô-đun) được kết nối với mỗi hook với thứ tự xác định. Mỗi mô-đun sẽ lần lượt được gọi và trả về một quyết định cho framework netfilter sau khi xử lý, cho biết cần phải làm gì với gói tin.

### 3. TCP/IP protocol stack

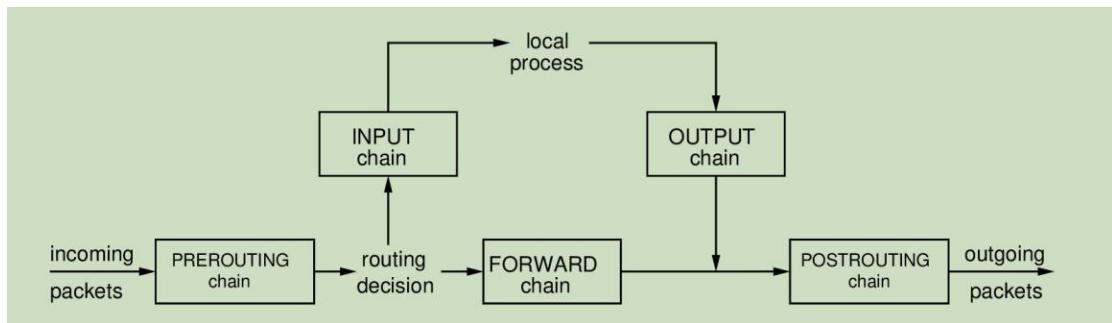
- **Tầng ứng dụng:** Đây là tầng mà các ứng dụng giao tiếp với nhau. Các ứng dụng như trình duyệt web, email, hoặc bất kỳ dịch vụ nào khác tạo ra dữ liệu cần gửi. Dữ liệu từ tầng ứng dụng được định dạng dưới dạng các thông điệp hoặc tệp, tùy thuộc vào ứng dụng.
- **Tầng vận chuyển:** dữ liệu từ ứng dụng sẽ được chia thành các phân đoạn nhỏ (segments) để dễ quản lý và truyền tải. Tầng này sử dụng các giao thức như TCP (đảm bảo kết nối và đáng tin cậy) hoặc UDP (không cần kết nối, không đảm bảo tính toàn vẹn dữ liệu). Với TCP, nó sẽ thiết lập kết nối giữa hai thiết bị bằng cách sử dụng quy trình 3-way handshake (bắt tay ba bước). TCP cũng đánh số và đảm bảo các phân đoạn dữ liệu đến đúng thứ tự. Sau đó, các data segment này sẽ được đóng gói với thông tin tiêu đề (header), bao gồm số cổng nguồn và đích, số thứ tự, ...
- **Tầng mạng:** Tầng này có nhiệm vụ định tuyến các gói tin qua các mạng khác nhau. Giao thức chính được sử dụng ở đây là IP (Internet Protocol). Các segment từ tầng giao vận sẽ được đóng gói thành các gói tin IP (IP packets). Trong gói tin IP, có các thông tin như địa chỉ IP nguồn và đích, giúp xác định đường đi của gói tin qua mạng.
- **Tầng liên kết dữ liệu:** Tầng này chịu trách nhiệm truyền các khung dữ liệu (frames) giữa các thiết bị trên cùng một mạng vật lý. Các gói tin IP từ tầng mạng sẽ được đóng gói thêm với thông tin của tầng liên kết dữ liệu, bao gồm địa chỉ MAC nguồn và đích.

- Tầng vật lý: Đây là tầng thấp nhất, nơi các khung dữ liệu (frames) được chuyển thành các tín hiệu điện hoặc tín hiệu quang để truyền qua cáp hoặc sóng. Tầng này chuyển đổi dữ liệu thành các bit và truyền qua phương tiện truyền dẫn như dây dẫn điện, cáp quang, hoặc qua không gian trong trường hợp mạng không dây.

## II. MÔ HÌNH



Kiến trúc Netfilter



Kiến trúc iptables

## III. TRIỂN KHAI

### 1. Netfilter

**Kịch bản 1: Hook LOCAL\_IN - Giới hạn số lượng kết nối SSH (Chỉ có 1 kết nối bất kỳ)**

Code C: [mã nguồn](#)

```

static struct nf_hook_ops *nf_hook_ex_ops = NULL;
#define MAX_SSH 1
static int ssh_count = 0;
static __be32 ssh_host_list[1];
static unsigned int nf_hook_ex(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
    struct iphdr *iph;
    struct tcphdr *tcp;
    bool found = false;

    if(!skb)
        return NF_ACCEPT;
    iph = ip_hdr(skb);

    if (iph->protocol == IPPROTO_TCP) {
        tcp = tcp_hdr(skb);
        if (ntohs(tcp->dest) == 22) {
            // Kiểm tra xem IP này đã tồn tại trong mảng chưa
            if (ssh_host_list[0] == iph->saddr) {
                found = true;
            }

            // Nếu IP chưa có trong mảng và đã đạt giới hạn thì chặn kết nối
            if (!found) {
                if (ssh_count >= MAX_SSH) {
                    printk(KERN_INFO "Maximum SSH hosts reached\n");
                    return NF_DROP;
                }
                else {
                    // Thêm IP mới vào mảng
                    ssh_host_list[0] = iph->saddr;
                    ++ssh_count;
                    printk(KERN_INFO "New SSH host allowed\n");
                }
            }
        }
    }
    return NF_ACCEPT;
}

```

Kịch bản 1: Hàm giới hạn số lượng ssh

Ý tưởng sẽ tạo ra một mảng có số lượng phần tử là 1. Phần tử này sẽ là 1 địa chỉ ip. Nếu như các gói tin ssh của địa chỉ này chưa nằm trong mảng và mảng đang rỗng, nghĩa là chưa có máy nào ssh vào host thì sẽ được chấp nhận kết nối và thêm vào mảng. Ngược lại các gói tin sẽ bị DROP.

Ta sẽ gắn nó ở hook LOCAL\_IN

```

static int __init kmod_init(void) {
    nf_hook_ex_ops = (struct nf_hook_ops*)kcalloc(1, sizeof(struct nf_hook_ops), GFP_KERNEL);
    if (nf_hook_ex_ops != NULL) {
        ssh_count = 0;
        /* đây là hàm callback `nf_hook_ex` kiểu nf_hookfn - định nghĩa trong include/linux/netfilter.h, line 47
         * - các tham số của hook mà người dùng định nghĩa phải khớp với kiểu nf_hookfn */
        nf_hook_ex_ops->hook = (nf_hookfn*)nf_hook_ex;

        /* Sự kiện mà hook này đăng ký */
        //nf_hook_ex_ops->hooknum = NF_INET_PRE_ROUTING;
        nf_hook_ex_ops->hooknum = NF_INET_LOCAL_IN;

        /* Chỉ xử lý các Internet (IPv4) packet */
        nf_hook_ex_ops->pf = NFPROTO_IPV4;

        /* Cài đặt độ ưu tiên của hook này ở mức độ cao nhất*/
        nf_hook_ex_ops->priority = NF_IP_PRI_FIRST;

        nf_register_net_hook(&init_net, nf_hook_ex_ops);
    }
    return 0;
}

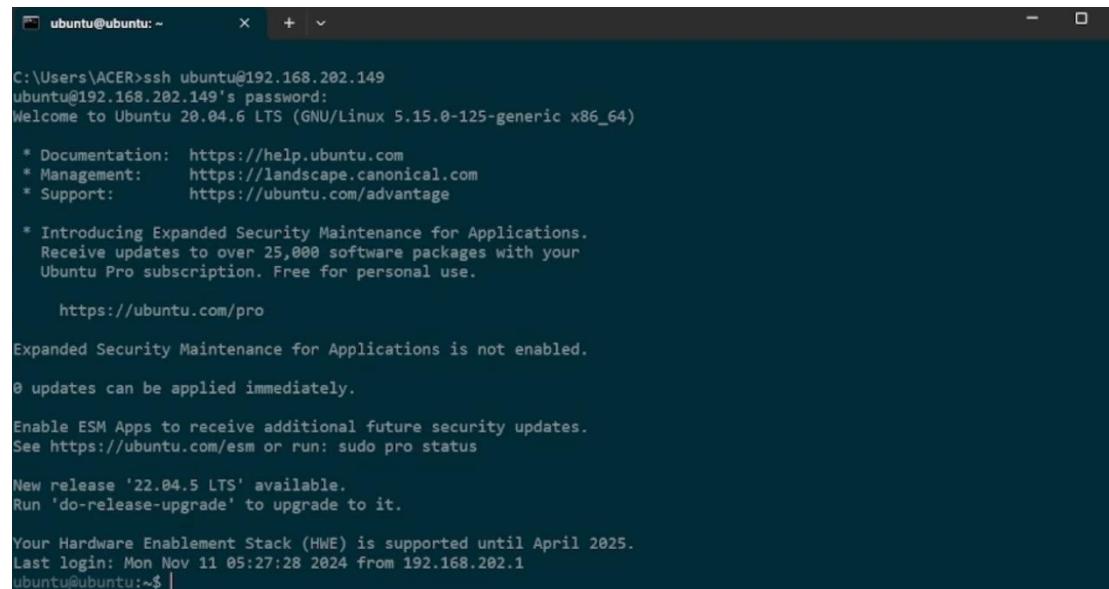
```

Kịch bản 1: Hook được gắn ở LOCAL\_IN

Khi insert mod lại thì ta sẽ reset biến count về 0. Ta gắn vào hook LOCAL\_IN thay vì PRE\_ROUTING vì LOCAL\_IN xử lý các gói tin đến được máy local, đã qua routing. Còn PRE\_ROUTING xử lý các gói tin đến trước khi routing gồm local và forwarding. Vì ta chỉ muốn giới hạn ssh cho máy local, tránh các xử lý không cần thiết với các gói tin forward qua máy.

Trước khi insert mode, ta thử ssh đến host

### Máy 1:



```
C:\Users\ACER>ssh ubuntu@192.168.202.149
ubuntu@192.168.202.149's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-125-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

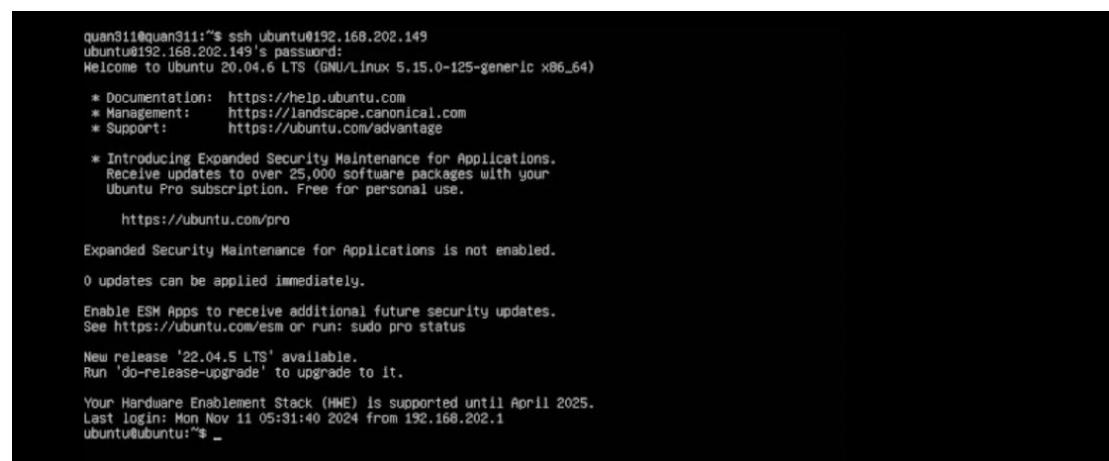
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Nov 11 05:27:28 2024 from 192.168.202.1
ubuntu@ubuntu:~$ |
```

Kịch bản 1: SSH tới host bằng Windows

### Máy 2



```
quan311@quan311:~$ ssh ubuntu@192.168.202.149
ubuntu@192.168.202.149's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-125-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

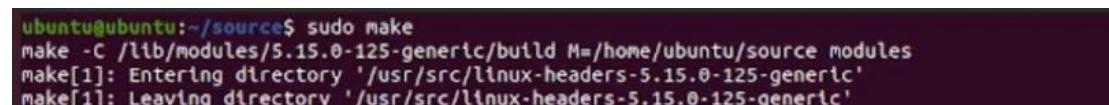
New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Nov 11 05:31:40 2024 from 192.168.202.1
ubuntu@ubuntu:~$ |
```

Kịch bản 1: SSH tới host bằng Ubuntu

Bây giờ ta tiến hành insert mode để chỉ giới hạn 1 máy ssh vào host

Tiến hành build



```
ubuntu@ubuntu:~/source$ sudo make
make -C /lib/modules/5.15.0-125-generic/build M=/home/ubuntu/source modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-125-generic'
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-125-generic'
```

### Kịch bản 1: Builld

Insert mode và đọc log.

Với máy đầu tiên ssh

```
C:\Users\ACER>ssh ubuntu@192.168.202.149
ubuntu@192.168.202.149's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-125-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Nov 11 05:31:52 2024 from 192.168.202.134
ubuntu@ubuntu:~$ |
```

[ 4903.225731] Exit  
[ 5182.871340] New SSH host allowed

Kịch bản 1: SSH từ máy Windows và đọc log

Máy thứ 2 ssh

```
[ 5213.840056] Maximum SSH hosts reached
[ 5214.870517] Maximum SSH hosts reached
[ 5216.887336] Maximum SSH hosts reached
[ 5221.142640] Maximum SSH hosts reached
[ 5229.335257] Maximum SSH hosts reached
[ 5245.463115] Maximum SSH hosts reached
```

Kịch bản 1: Log của máy Ubuntu ssh tới host

Link demo: [here](#)

### Kịch bản 2: Chặn các máy muốn ping hoặc telnet đến một host cụ thể (10.0.3.8)

Code C: [mã nguồn](#)

Ý tưởng giải quyết:

Bước 1: Xác định gói tin cần chặn

Gói tin ICMP:

- Kiểm tra nếu giao thức của gói tin là ICMP (IPPROTO\_ICMP)
- Nếu gói tin là kiểu ICMP\_ECHO (ping request) và địa chỉ đích là 10.0.3.8, thì chặn

Gói tin TCP đến cổng 23:

- Kiểm tra nếu giao thức của gói tin là TCP (IPPROTO\_TCP)
- Nếu địa chỉ đích là 10.0.3.8 và cổng đích là 23 (Telnet), thì chặn

Bước 2: Xây dựng hàm chặn gói tin

Viết các hàm để xử lý từng loại gói tin cần chặn:

BlockICMP: Chặn gói tin ICMP ping

```
// Hàm chặn các gói ICMP ping tới địa chỉ IP 10.0.3.8
unsigned int blockICMP(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
    struct iphdr *iph;          // Header IP
    struct icmphdr *icmph;     // Header ICMP

    char ip[16] = "10.0.3.8"; // Địa chỉ IP cần chặn
    u32 ip_addr;               // Địa chỉ IP dạng nhị phân

    // Kiểm tra gói tin có hợp lệ không
    if (!skb)
        return NF_ACCEPT;

    iph = ip_hdr(skb); // Truy xuất header IP từ gói tin
    // Chuyển đổi địa chỉ IP từ chuỗi dạng "10.0.3.8" sang nhị phân
    in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);

    // Kiểm tra giao thức ICMP
    if (iph->protocol == IPPROTO_ICMP)
    {
        icmph = icmp_hdr(skb); // Truy xuất header ICMP từ gói tin
        // Nếu gói tin là ping (ICMP_ECHO) và gửi đến địa chỉ 10.0.3.8, thì chặn
        if (iph->daddr == ip_addr && icmph->type == ICMP_ECHO)
        {
            printk(KERN_WARNING "*** Dropping %pI4 (ICMP) \n", &(iph->daddr));
            return NF_DROP; // Chặn gói tin
        }
    }
    return NF_ACCEPT; // Cho phép gói tin nếu không khớp điều kiện
}
```

Kịch bản 2: Hàm block ICMP

BlockTelnet: Chặn gói tin TCP đến cổng 23 (Telnet)

```

// Hàm chặn các gói TCP Telnet tới cổng 23 của địa chỉ 10.0.3.8
unsigned int blockTelnet(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
    struct iphdr *iph; // Header IP
    struct tcphdr *tcph; // Header TCP

    u16 port = 23; // Cổng Telnet
    char ip[16] = "10.0.3.8"; // Địa chỉ IP cần chặn
    u32 ip_addr; // Địa chỉ IP dạng nhị phân

    if (!skb)
        return NF_ACCEPT;

    iph = ip_hdr(skb); // Truy xuất header IP từ gói tin
    in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);

    // Kiểm tra giao thức TCP
    if (iph->protocol == IPPROTO_TCP)
    {
        tcph = tcp_hdr(skb); // Truy xuất header TCP từ gói tin
        // Nếu gói tin đến địa chỉ 10.0.3.8 và cổng đích là 23, thì chặn
        if (iph->daddr == ip_addr && ntohs(tcph->dest) == port)
        {
            printk(KERN_WARNING "**** Dropping %pI4 (TCP), port %d\n", &(iph->daddr), port);
            return NF_DROP; // Chặn gói tin
        }
    }
    return NF_ACCEPT; // Cho phép gói tin nếu không khớp điều kiện
}

```

Kịch bản 2: Hàm block Telnet

### Bước 3: Sử dụng Netfilter để chặn gói tin

- Dùng Netfilter hooks để can thiệp vào đường đi của gói tin qua ngăn xếp mạng
- Các hook sẽ được đăng ký vào điểm LOCAL\_OUT để kiểm tra gói tin trước khi rời hệ thống

```

// Đăng ký hook chặn gói ICMP
hook2.hook = blockICMP;
hook2.hooknum = NF_INET_LOCAL_OUT; // Hook tại LOCAL_OUT
hook2(pf = PF_INET;
hook2.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook2);

// Đăng ký hook chặn gói Telnet
hook3.hook = blockTelnet;
hook3.hooknum = NF_INET_LOCAL_OUT; // Hook tại LOCAL_OUT
hook3(pf = PF_INET;
hook3.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook3);

```

Kịch bản 2: Các hook được đăng ký

### Bước 4: In thông tin ra log để kiểm tra hoạt động

- Viết một hook để in ra thông tin gói tin nhằm kiểm tra hook hoạt động đúng
- Hiển thị thông tin như: nguồn, đích, giao thức, và điểm hook

```
// Hàm in thông tin gói tin đi qua các hook
unsigned int printInfo(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
    struct iphdr *iph;
    char *hook;
    char *protocol;

    // Xác định hook hiện tại (điểm trên đường đi của gói tin)
    switch (state->hook)
    {
        case NF_INET_LOCAL_IN:
            hook = "LOCAL_IN";
            break;
        case NF_INET_LOCAL_OUT:
            hook = "LOCAL_OUT";
            break;
        case NF_INET_PRE_ROUTING:
            hook = "PRE_ROUTING";
            break;
        case NF_INET_POST_ROUTING:
            hook = "POST_ROUTING";
            break;
        case NF_INET_FORWARD:
            hook = "FORWARD";
            break;
        default:
            hook = "IMPOSSIBLE";
            break;
    }
    printk(KERN_INFO "*** %s\n", hook); // In thông tin hook

    iph = ip_hdr(skb); // Truy xuất header IP từ gói tin
    // Xác định giao thức
    switch (iph->protocol)
    {
        case IPPROTO_TCP:
            protocol = "TCP";
            break;
        case IPPROTO_ICMP:
            protocol = "ICMP";
            break;
        default:
            protocol = "OTHER";
            break;
    }
    // In địa chỉ nguồn, đích và giao thức của gói tin
    printk(KERN_INFO " %pI4 --> %pI4 (%s)\n", &(iph->saddr), &(iph->daddr), protocol);

    return NF_ACCEPT; // Cho phép gói tin đi qua
}
```

Kịch bản 2: Hàm printInfo

Bước 5: Khởi tạo và hủy bỏ module

Đăng ký và hủy các hook trong hàm registerFilter và removeFilter

```

// Hàm khởi tạo module và đăng ký các bộ lọc
int registerFilter(void)
{
    printk(KERN_INFO "Registering filters.\n");

    // Đăng ký hook in thông tin gói tin
    hook1.hook = printInfo;
    hook1.hooknum = NF_INET_LOCAL_OUT; // Hook tại LOCAL_OUT
    hook1(pf) = PF_INET;
    hook1.priority = NF_IP_PRI_FIRST; // Ưu tiên cao nhất
    nf_register_net_hook(&init_net, &hook1);

    // Đăng ký hook chặn gói ICMP
    hook2.hook = blockICMP;
    hook2.hooknum = NF_INET_LOCAL_OUT; // Hook tại LOCAL_OUT
    hook2(pf) = PF_INET;
    hook2.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &hook2);

    // Đăng ký hook chặn gói Telnet
    hook3.hook = blockTelnet;
    hook3.hooknum = NF_INET_LOCAL_OUT; // Hook tại LOCAL_OUT
    hook3(pf) = PF_INET;
    hook3.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &hook3);

    return 0;
}

// Hàm gỡ bỏ module và hủy đăng ký các bộ lọc
void removeFilter(void)
{
    printk(KERN_INFO "The filters are being removed. \n");
    nf_unregister_net_hook(&init_net, &hook1); // Hủy hook in thông tin
    nf_unregister_net_hook(&init_net, &hook2); // Hủy hook chặn ICMP
    nf_unregister_net_hook(&init_net, &hook3); // Hủy hook chặn Telnet
}

// Định nghĩa các macro để chỉ định hàm khởi tạo và hủy bỏ module
module_init(registerFilter);
module_exit(removeFilter);

// Thông tin về giấy phép
MODULE_LICENSE("GPL");

```

Kịch bản 2: Hàm registerFilter và hàm removeFilter

Triển khai:

Trước khi áp dụng Netfilter module, ta thực hiện ping và telnet thành công đến máy 10.0.3.8

```

Ubuntu x Host x
Activities Terminal Dec 4 18:11
host@ubuntu: ~

thaont@ubuntu:~$ ping 10.0.3.8
PING 10.0.3.8 (10.0.3.8) 56(84) bytes of data.
64 bytes from 10.0.3.8: icmp_seq=1 ttl=64 time=1.68 ms
64 bytes from 10.0.3.8: icmp_seq=2 ttl=64 time=0.469 ms
64 bytes from 10.0.3.8: icmp_seq=3 ttl=64 time=0.443 ms
64 bytes from 10.0.3.8: icmp_seq=4 ttl=64 time=0.318 ms
64 bytes from 10.0.3.8: icmp_seq=5 ttl=64 time=0.443 ms
64 bytes from 10.0.3.8: icmp_seq=6 ttl=64 time=0.443 ms
64 bytes from 10.0.3.8: icmp_seq=7 ttl=64 time=0.443 ms
64 bytes from 10.0.3.8: icmp_seq=8 ttl=64 time=0.447 ms
64 bytes from 10.0.3.8: icmp_seq=9 ttl=64 time=0.729 ms
...
-- 10.0.3.8 ping statistics --
9 packets transmitted, 9 received, 0% packet loss, time 8157ms
rtt min/avg/max/mdev = 0.316/0.586/1.682/0.403 ms
thaont@ubuntu:~$ telnet 10.0.3.8
Trying 10.0.3.8...
Connected to 10.0.3.8.
Escape character is '^]'.
Ubuntu 20.04.6 LTS
ubuntu login: host
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-126-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Thu Dec  5 09:10:17 +07 2024 from 10.0.3.5 on pts/0
host@ubuntu:~$
```

Kịch bản 2: Ping và telnet đến 10.0.3.8 trước khi áp dụng Netfilter

Ta cần chuẩn bị hai file là Block\_to\_host.c theo như ý tưởng đã trình bày và Makefile được dùng để build file. Tiếp theo, ta thực hiện makefile bằng lệnh make, sau đó dùng lệnh insmod để cài đặt Netfilter module vào kernel và lsmod để kiểm tra

```

Ubuntu x Host x
Activities Terminal Dec 4 18:27
thaont@ubuntu:~/NT131$ ls
Block_to_host.c Makefile
thaont@ubuntu:~/NT131$ sudo make
[sudo] password for thaont:
make -C /lib/modules/5.15.0-126-generic/build M=/home/thaont/NT131 modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-126-generic'
  CC [M] /home/thaont/NT131/Block_to_host.o
  MODPOST /home/thaont/NT131/Module.symvers
  CC [M] /home/thaont/NT131/Block_to_host.mod.o
  LD [M] /home/thaont/NT131/Block_to_host.ko
  BTF [M] /home/thaont/NT131/Block_to_host.ko
  Skipping BTF generation for /home/thaont/NT131/Block_to_host.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-126-generic'
thaont@ubuntu:~/NT131$ sudo insmod Block_to_host.ko
thaont@ubuntu:~/NT131$ lsmod | grep Block_to_host
Block_to_host    16384  0
thaont@ubuntu:~/NT131$
```

Kịch bản 2: Build và cài đặt Netfilter module vào kernel

Sau khi đã thực hiện xong, ta sẽ thử ping đến 10.0.3.8 thì ta thấy bị mất 100% và sử dụng dmesg để kiểm tra log, ta thấy các gói tin ICMP đến 10.0.3.8 đều bị DROP.

Kịch bản 2: Thực hiện ping đến 10.0.3.8 sau khi áp dụng Netfilter

Tương tự với telnet cũng vậy, ta thấy được việc kết nối đã bị chặn và log hiển thị các gói tin TCP đến 10.0.3.8 port 23 đều bị DROP.



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "thaont@ubuntu: ~/NT131". The terminal content shows the following:

```
thaont@ubuntu:~$ netcat -l -p 23 > /tmp/nc1
[1711.256055] *** Dropping 10.0.3.8 (TCP), port 23
[1715.352359] *** Dropping 10.0.3.8 (TCP), port 23
[1719.411139] *** LOCAL_OUT
[1719.411143] 10.0.3.5 --> 10.0.3.2 (OTHER)
[1723.543483] *** Dropping 10.0.3.8 (TCP), port 23
[1739.669977] *** Dropping 10.0.3.8 (TCP), port 23
[1747.871168] *** LOCAL_OUT
[1747.871171] 127.0.0.1 --> 127.0.0.1 (TCP)
[1747.871187] *** LOCAL_OUT
[1747.871187] 127.0.0.1 --> 127.0.0.1 (TCP)
thaont@ubuntu:~$
```

Kích bản 2: Thực hiện telnet đến 10.0.3.8 sau khi áp dụng Netfilter

Link demo: [here](#)

**Kịch bản 3: Chặn các gói tin từ một địa chỉ IP nhất định khi cố gắng truy cập Apache port 80 trên máy chủ.**

## Code C: mã nguồn

#### Bước 1: Xác định gói tin cần chẩn

- Gói tin TCP đến cổng 80
  - Kiểm tra nếu giao thức của gói tin là TCP (IPPROTO\_TCP)
  - Nếu địa chỉ IP của máy client là 192.168.17.132

```
// IP của client cần chặn (192.168.17.132)
#define BLOCKED_IP "192.168.17.132"
#define BLOCKED_PORT 80
```

Kịch bản 3: Thiết lập địa chỉ ip và port mục tiêu

## Bước 2: Xây dựng hàm chặn gói tin

```
// Hàm callback xử lý gói tin
static unsigned int block_client_func(void *priv, struct sk_buff *skb, const struct nf_hook_state *state) {
    struct iphdr *ip_header;
    struct tcphdr *tcp_header;
    unsigned int src_ip;

    // Kiểm tra gói tin
    if (!skb) return NF_ACCEPT;

    ip_header = ip_hdr(skb); // Lấy header IP
    if (!ip_header) return NF_ACCEPT;

    // Kiểm tra giao thức (chỉ xử lý TCP)
    if (ip_header->protocol != IPPROTO_TCP) return NF_ACCEPT;

    tcp_header = tcp_hdr(skb); // Lấy header TCP
    if (!tcp_header) return NF_ACCEPT;

    // Chuyển đổi IP nguồn từ dạng chuỗi sang số
    src_ip = in_aton(BLOCKED_IP);

    // Kiểm tra nếu IP nguồn và cổng đích khớp
    if (ip_header->saddr == src_ip && ntohs(tcp_header->dest) == BLOCKED_PORT) {
        printk(KERN_INFO "Blocked IP: %pI4 trying to access Apache on port %d\n",
               &ip_header->saddr, BLOCKED_PORT);
        return NF_DROP; // Chặn gói tin
    }

    return NF_ACCEPT; // Cho phép gói tin khác
}
```

Kịch bản 3: Hàm chặn gói tin

- block\_client\_func: hàm callback được gọi mỗi khi gói tin đến skb: là con trỏ trả về cho hàm nf\_register()
- ip\_header = ip\_hdr(skb): Lấy địa chỉ ip của gói tin
- ip\_header-> protocol != IPPROTO\_TCP: kiểm tra giao thức của gói tin, nếu không phải TCP thì cho qua.
- ip\_header->saddr == src\_ip: kiểm tra xem có phải địa chỉ IP của client(192.168.17.132) không.
- ntohs(tcp\_header->dest): là cổng đích của gói tin, hàm ntohs để lấy ra port đích của gói tin.
- Kết hợp ip và port đích của gói tin nếu đúng yêu cầu thì chặn gói tin, còn nếu không thì return NF\_ACCEPT;
- printk(KERN\_INFO) để in ra log kiểm tra hoạt động

## Bước 3: Viết hàm khởi tạo và hủy bỏ module

```

// Hàm khởi tạo module
static int __init netfilter_block_client_init(void) {
    printk(KERN_INFO "Netfilter Block Client Module Loaded.\n");

    // Cấu hình hook
    netfilter_hook.hook = block_client_func;
    netfilter_hook.hooknum = NF_INET_PRE_ROUTING;
    netfilter_hook.pf = PF_INET;
    netfilter_hook.priority = NF_IP_PRI_FIRST;

    // Đăng ký hook
    nf_register_net_hook(&init_net, &netfilter_hook);

    return 0;
}

// Hàm gỡ bỏ module
static void __exit netfilter_block_client_exit(void) {
    printk(KERN_INFO "Netfilter Block Client Module Unloaded.\n"

    // Hủy đăng ký hook
    nf_unregister_net_hook(&init_net, &netfilter_hook);
}

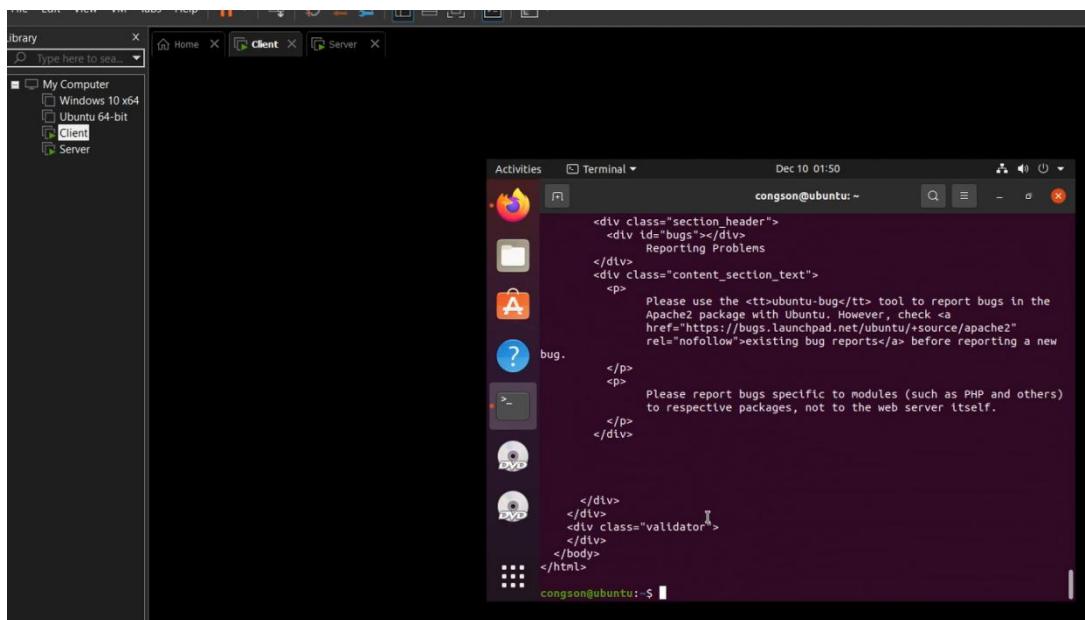
module_init(netfilter_block_client_init);
module_exit(netfilter_block_client_exit);

```

Kịch bản 3: Hàm khởi tạo và hủy bỏ module

#### Bước 4: Triển khai thí nghiệm

Trước khi nạp module, sử dụng lệnh curl http://192.168.17.133 (IP Server) để gửi một yêu cầu HTTP đến máy server.



Kịch bản 3: Gửi yêu cầu HTTP đến host

Biên dịch và nạp module vào kernel.

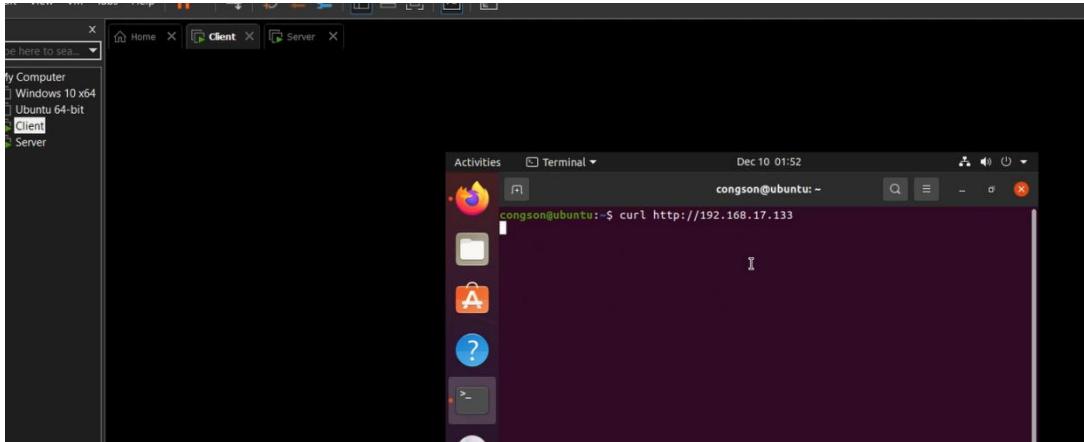
```

congson@ubuntu:~/CongSonReport$ sudo make
make -C /lib/modules/5.15.0-126-generic/build M=/home/congson/CongSonReport modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-126-generic'
  CC [M] /home/congson/CongSonReport/netfilter_block_client.o
  MODPOST /home/congson/CongSonReport/module.symvers
  CC [M] /home/congson/CongSonReport/netfilter_block_client.mod.o
  LD [M] /home/congson/CongSonReport/netfilter_block_client.ko
  BTF [M] /home/congson/CongSonReport/netfilter_block_client.ko
Skipping BTF generation for /home/congson/CongSonReport/netfilter_block_client.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-126-generic'
congson@ubuntu:~/CongSonReport$ ls
Module.symvers  netfilter_block_client.ko  netfilter_block_client.mod.c  netfilter_block_client.o
modules.order  netfilter_block_client.c  netfilter_block_client.mod  netfilter_block_client.mod.o
congson@ubuntu:~/CongSonReport$ sudo insmod netfilter_block_client.ko
congson@ubuntu:~/CongSonReport$ 

```

Kịch bản 3: Biên dịch và nạp module

Máy client để gửi yêu cầu HTTP đến server và nó đã không phản hồi



Kịch bản 3: Gửi yêu cầu sau khi nạp module

Dùng lệnh dmesg | tail để xem log và thí nghiệm thành công

```

modules.order  netfilter_block_client netfilter_block_client.mod.o  netfilter_block_client.ko
congson@ubuntu:~/CongSonReport$ sudo insmod netfilter_block_client.ko
congson@ubuntu:~/CongSonReport$ dmesg | tail
[ 4165.981588] audit: type=1400 audit(1733824278.394:60): apparmor="STATUS" operation="profile_replace" info
-store,ubuntu-software" pid=61209 comm="apparmor_parser"
[ 4166.018438] audit: type=1400 audit(1733824278.433:61): apparmor="STATUS" operation="profile_replace" info
-store,ubuntu-software-local-file" pid=61210 comm="apparmor_parser"
[ 4190.515189] Netfilter Block Client Module Loaded.
[ 4202.212090] systemd-rc-local-generator[63768]: /etc/rc.local is not marked executable, skipping.
[ 4202.506601] systemd-rc-local-generator[63794]: /etc/rc.local is not marked executable, skipping.
[ 4221.075210] Blocked IP: 192.168.17.132 trying to access Apache on port 80
[ 4222.089888] Blocked IP: 192.168.17.132 trying to access Apache on port 80
[ 4224.104868] Blocked IP: 192.168.17.132 trying to access Apache on port 80
[ 4228.265435] Blocked IP: 192.168.17.132 trying to access Apache on port 80
[ 4236.458038] Blocked IP: 192.168.17.132 trying to access Apache on port 80
congson@ubuntu:~/CongSonReport$ sudo rmmod netfilter_block_client.ko

```

Kịch bản 3: Xem log

Link demo: [here](#)

## 2. Iptables

### Kịch bản 4: OUTPUT - Chặn truy cập Internet

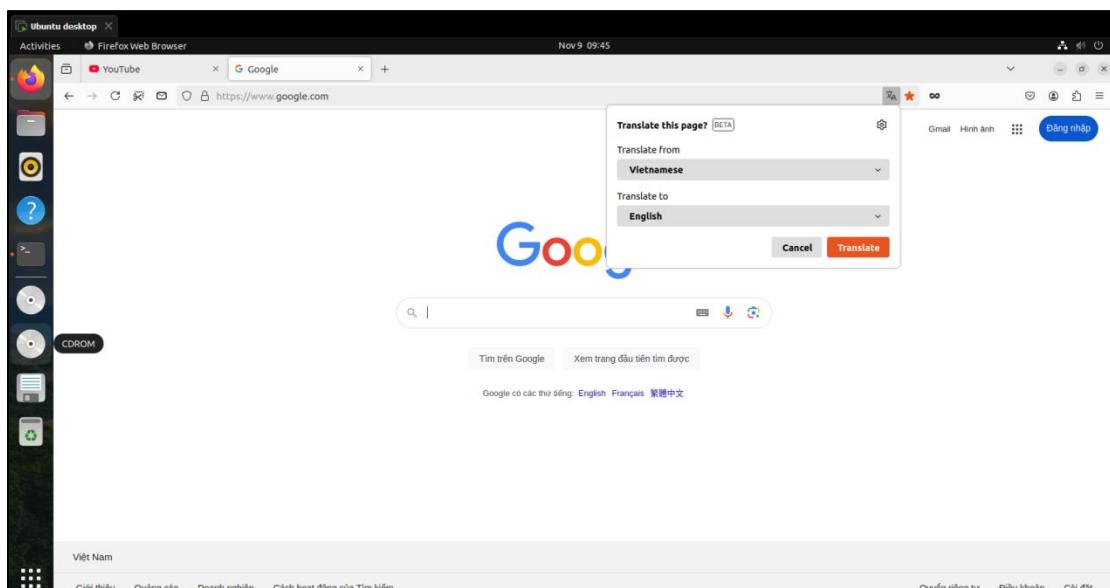
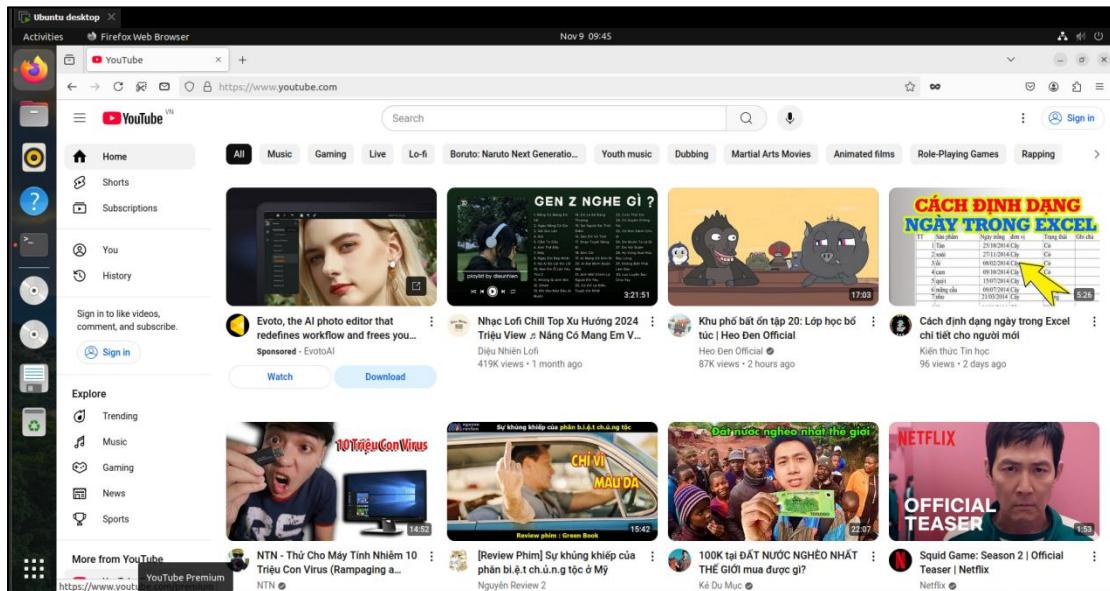
Ban đầu ta kiểm tra các rule và truy cập một vài website như Google, youtube

```

quantc@quantc-ubuntu: ~$ sudo iptables -L --line-number
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
quantc@quantc-ubuntu: ~$ 

```

Kịch bản 4: Kiểm tra các rules của iptables



Kịch bản 4: Kiểm tra truy cập Internet trước khi có rules

Bây giờ ta sẽ áp dụng rule vào chain OUPUT để REJECT các gói tin đi ra ngoài với destination port là 443 (HTTPS)

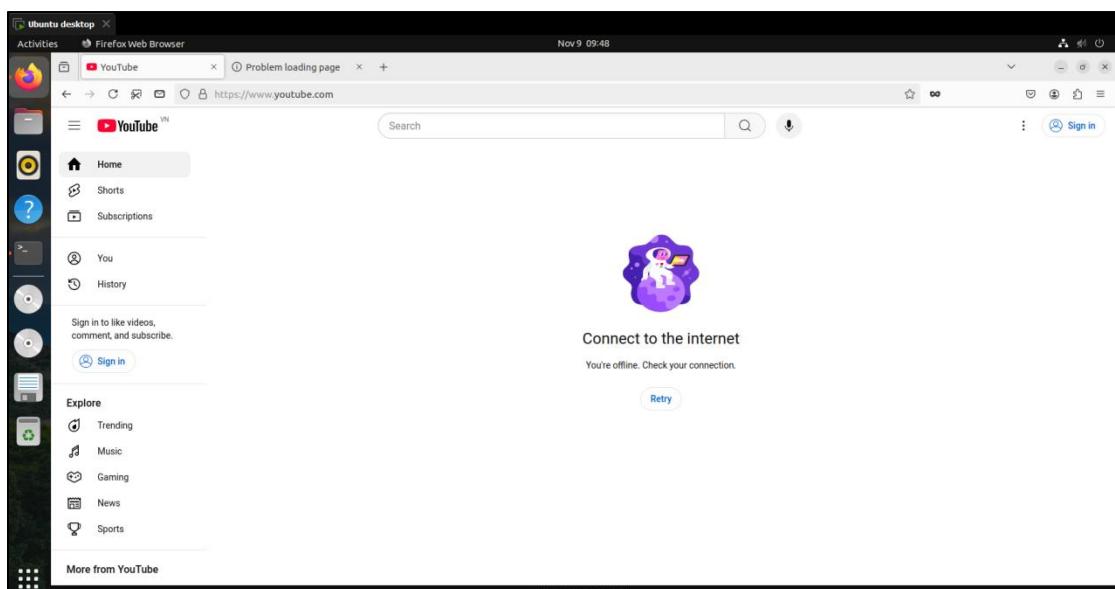
```

Ubuntu desktop
Activities Terminal Nov 9 09:47
quantc@quantc-ubuntu:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -j REJECT
quantc@quantc-ubuntu:~$ sudo iptables -A OUTPUT -p udp --dport 443 -j REJECT
quantc@quantc-ubuntu:~$ sudo iptables -L --line-number
Chain INPUT (policy ACCEPT)
num  target     prot opt source         destination
  1  ACCEPT     all  --  anywhere       anywhere
Chain FORWARD (policy ACCEPT)
num  target     prot opt source         destination
  1  ACCEPT     all  --  anywhere       anywhere
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source         destination
  1  REJECT    tcp  --  anywhere      anywhere          tcp dpt:https reject-with icmp-port-unreachable
  2  REJECT    udp  --  anywhere      anywhere          udp dpt:https reject-with icmp-port-unreachable
quantc@quantc-ubuntu:~$ 

```

Kịch bản 4: Áp dụng rules

Bởi vì youtube có sử dụng QUIC (UDP) nên ta chặn thêm giao thức này. Kết quả:



Kịch bản 4: Kiểm tra truy cập sau khi áp dụng rules

Mặc dù có kết nối internet nhưng vẫn không truy cập được

Link demo: [here](#)

### Kịch bản 5: INPUT - Ghi log và gửi cảnh báo về mail khi bị scanning port

Đầu tiên cần tải logwatch và postfix

`sudo apt install logwatch`

`sudo apt install postfix`

Tiếp đó thêm rule iptables

```

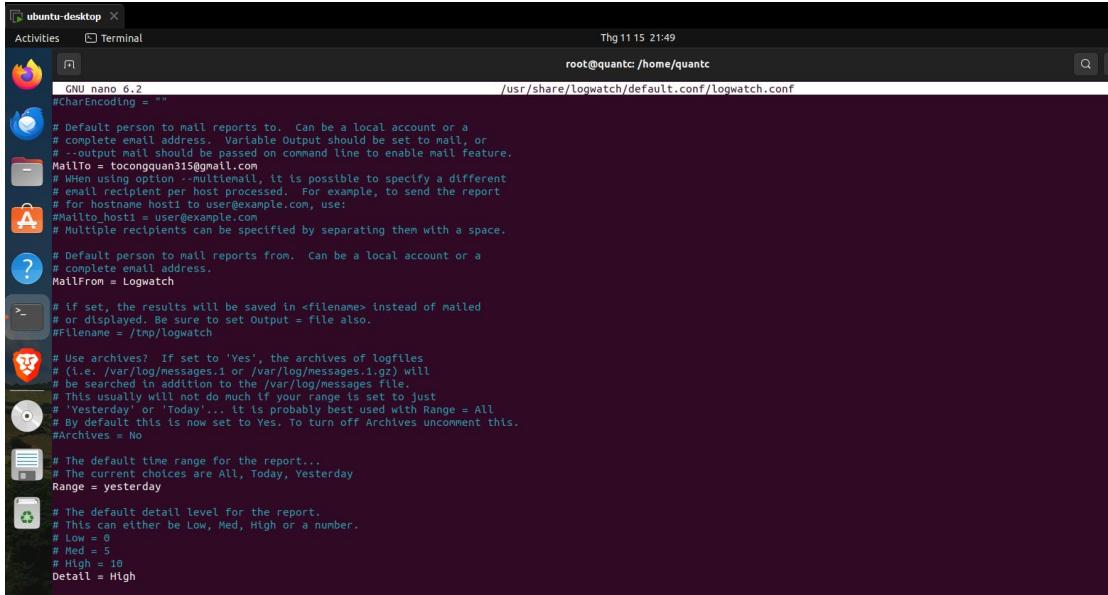
ubuntu-desktop
Activities Terminal Thg 11 15 21:47
root@quantc:/home/quantc#
root@quantc:/home/quantc# iptables -A INPUT -p tcp --syn -m limit --limit 5/minute --limit-burst 10 -j LOG --log-prefix "Port scan detected: " --log-level 4
root@quantc:/home/quantc# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source         destination
LOG       tcp  --  anywhere      anywhere          tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 5/min burst 10 LOG level warning prefix "Port scan detected: "
Chain FORWARD (policy ACCEPT)
target     prot opt source         destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source         destination
root@quantc:/home/quantc# 

```

#### Kịch bản 5: Thêm rules iptables

Thêm vào chain INPUT, với giao thức là TCP, chỉ xét những gói SYN khi bắt đầu kết nối. Đặt limit là ghi log tối đa 5 lần trên một phút. Đặt một burst là 10 để nếu có một lưu lượng đột ngột lớn hơn limit khi bắt đầu thì ta sẽ có thể ghi được log đó tối đa 10 log. Để target là LOG để ghi log và đặt level 4 (Warning).

#### Chỉnh sửa email nhận là email của mình



```
GNU nano 6.2
#CharEncoding = ""

# Default person to mail reports to. Can be a local account or a
# complete email address. Variable Output should be set to mail, or
# -output mail should be passed on command line to enable mail feature.
Mailto = tocongquan315@gmail.com
# When using option --multimail, it is possible to specify a different
# email recipient per host processed. For example, to send the report
# for hostname host1 to user@example.com, use:
#Mailto.host1 = user@example.com
# Multiple recipients can be specified by separating them with a space.

# Default person to mail reports from. Can be a local account or a
# complete email address.
MailFrom = Logwatch

# if set, the results will be saved in <filename> instead of mailed
# or displayed. Be sure to set Output = file also.
#Filename = /tmp/logwatch

# Use archives? If set to 'Yes', the archives of logfiles
# (i.e. /var/log/messages.1 or /var/log/messages.1.gz) will
# be searched in addition to the /var/log/messages file
# This usually will not do much if your range is set to just
# 'Yesterday' or 'Today' as it is probably best used with Range = All
# By default this is now set to Yes. To turn off Archives uncomment this.
#Archives = No

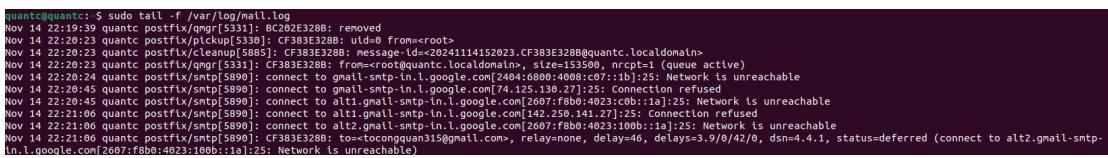
# The default time range for the report...
# The current choices are All, Today, Yesterday
Range = yesterday

# The default detail level for the report.
# This can either be Low, Med, High or a number.
# Low = 0
# Med = 5
# High = 10
Detail = High
```

#### Kịch bản 5: Chính sửa email nhận

Đặt MailTo đến email bản thân, range đặt yesterday để gửi nhưng log của ngày hôm qua, detail đặt mức High để nhận đầy đủ thông tin

Ngay lúc này khi ta giả sử gửi log đó qua mail khi thực hiện scanning thì sẽ báo lỗi



```
quantc@quantc: $ sudo tail -f /var/log/mail.log
Nov 14 22:19:39 quantc postfix/pickup[5331]: BC2D0E32B8: removed
Nov 14 22:20:23 quantc postfix/pickup[5330]: CF383E32B8: uid=0 from=<root>
Nov 14 22:20:23 quantc postfix/cleanup[5330]: CF383E32B8@quantc.localdomain>
Nov 14 22:20:23 quantc postfix/qmgr[5331]: CF383E32B8: message-id=<>202411152023.CF383E32B8@quantc.localdomain>
Nov 14 22:20:23 quantc postfix/smtp[5331]: to=quantc@quantc.localdomain, size=153500, nrcpt=1 (queue active)
Nov 14 22:20:45 quantc postfix/smtp[5331]: connect to gmail-smtp-in.l.google.com[24.125.100.25]:25: Network is unreachable
Nov 14 22:20:45 quantc postfix/smtp[5330]: connect to alt1.gmail-smtp-in.l.google.com[2667:f8b0:4023:c0b1:1a1]:25: Connection refused
Nov 14 22:21:06 quantc postfix/smtp[5330]: connect to alt1.gmail-smtp-in.l.google.com[2667:f8b0:4023:c0b1:1a1]:25: Network is unreachable
Nov 14 22:21:06 quantc postfix/smtp[5331]: connect to alt2.gmail-smtp-in.l.google.com[2667:f8b0:4023:100b::1a1]:25: Network is unreachable
Nov 14 22:21:06 quantc postfix/smtp[5331]: to=tocongquan315@gmail.com, relay=none, delay=46, delays=3.9/0/42/0, dsn=4.1.1, status=deferred (connect to alt2.gmail-smtp-in.l.google.com[2667:f8b0:4023:100b::1a1]:25: Network is unreachable)
```

#### Kịch bản 5: Bị chặn port 25

Lí do vì cổng 25 thường bị nhà cung cấp chặn để tránh việc spam. Do đó ta sẽ thiết lập cho phép trên cổng 587 (TSL)

```

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 3.6 on
# fresh installs.
compatibility_level = 3.6

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
#smtpd_tls_security_level=may
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache

#add
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_security_level = encrypt
smtp_tls_CACert = /etc/ssl/certs/ca-certificates.crt
myhostname = quantc.localdomain
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, quantc, localhost.localdomain, , localhost
#relayhost
mynetworks = 127.0.0.0/8 [:ffff:127.0.0.0]/104 [:1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

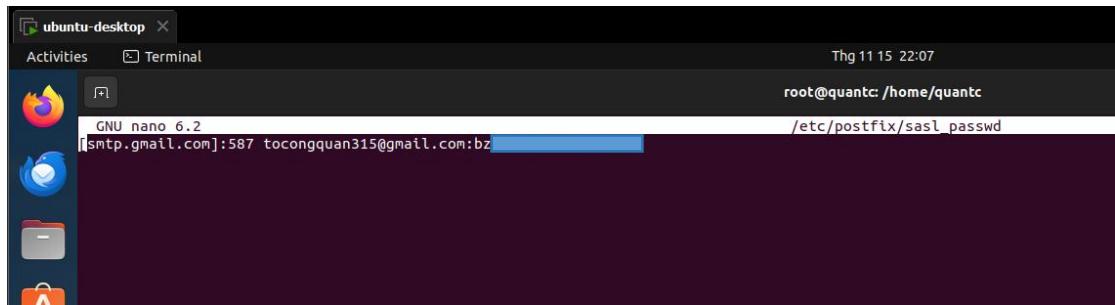
```

Kịch bản 5: Thiết lập cho phép port 587

Thêm các dòng #add.

- relayhost = [smtp.gmail.com]:587 -> Chỉ định máy chủ SMTP mà Postfix sẽ sử dụng để gửi email ra ngoài
- smtp\_sasl\_auth\_enable = yes -> Khi Postfix kết nối đến smtp.gmail.com, nó sẽ sử dụng thông tin xác thực từ file /etc/postfix/sasl\_passwd để đăng nhập.
- smtp\_sasl\_password\_maps = hash:/etc/postfix/sasl\_passwd -> Chỉ định đường dẫn đến file chứa thông tin tài khoản và mật khẩu dùng cho xác thực SASL.
- smtp\_sasl\_security\_options = noanonymous -> Đảm bảo rằng Postfix chỉ gửi thông tin xác thực hợp lệ đến Gmail.
- smtp\_tls\_security\_level = encrypt -> Khi Postfix gửi email, nó bắt buộc mã hóa phiên làm việc SMTP giữa máy chủ của bạn và máy chủ Gmail để bảo vệ dữ liệu.
- smtp\_tls\_CAfile = /etc/ssl/certs/ca-certificates.crt -> Khi Postfix kết nối đến smtp.gmail.com, nó kiểm tra chứng chỉ SSL/TLS của Gmail có hợp lệ hay không bằng cách so sánh với các chứng chỉ trong file này.

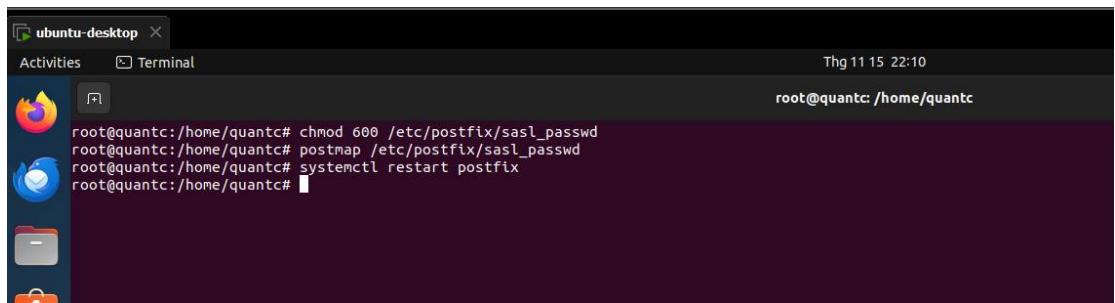
Tiếp đó, thiết lập tài khoản email gồm địa chỉ và mật khẩu ứng dụng. Vì email có xác thực 2 yếu tố, nên thay vì mật khẩu của email thì sẽ đặt mật khẩu ứng dụng



```
ubuntu-desktop
Activities Terminal
GNU nano 6.2
[sntp.gmail.com]:587 tocongquan315@gmail.com:bz
root@quantc: /home/quantc
/etc/postfix/sasl_passwd
```

Kịch bản 5: Thiết lập email và mật khẩu

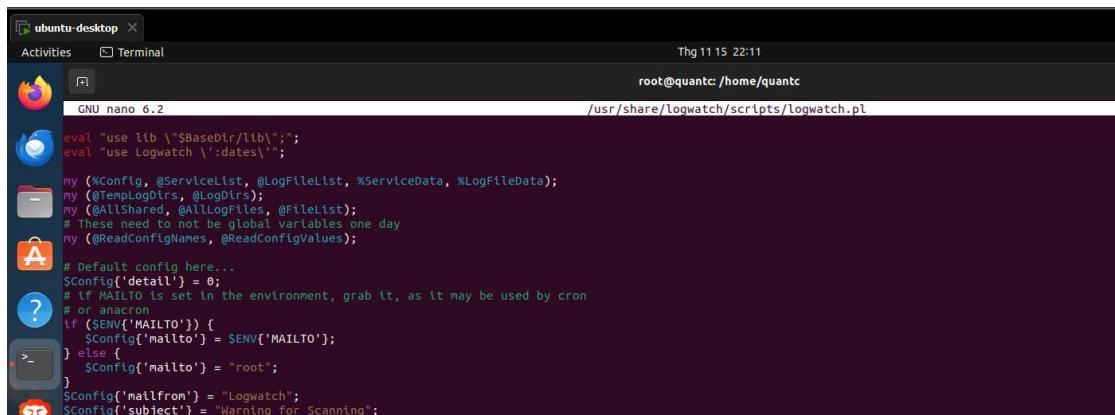
Cuối cùng là cấp quyền bảo mật và khởi động lại dịch vụ



```
ubuntu-desktop
Activities Terminal
root@quantc: /home/quantc# chmod 600 /etc/postfix/sasl_passwd
root@quantc: /home/quantc# postmap /etc/postfix/sasl_passwd
root@quantc: /home/quantc# systemctl restart postfix
root@quantc: /home/quantc#
```

Kịch bản 5: Cấp quyền bảo mật và khởi động lại

Có thể thay đổi tiêu đề gửi mail để biết là mail đó để làm gì



```
ubuntu-desktop
Activities Terminal
GNU nano 6.2
root@quantc: /home/quantc
/usr/share/logwatch/scripts/logwatch.pl

eval "use lib '$BaseDir/lib';";
eval "use Logwatch 'dates';"

my (@Config, @ServiceList, @LogFileList, %ServiceData, %LogFileData);
my (@TempLogdirs, @Logdirs);
my (@AllShared, @AllLogFiles, @FileList);
# These need to not be global variables one day
my (@ReadConfigNames, @ReadConfigValues);

# Default config here...
$Config{'detail'} = 0;
# If MAILTO is set in the environment, grab it, as it may be used by cron
# or anacron
if ($ENV{'MAILTO'}) {
    $Config{'mailto'} = $ENV{'MAILTO'};
} else {
    $Config{'mailto'} = "root";
}
$Config{'mailfrom'} = "Logwatch";
$Config{'subject'} = "Warning for Scanning";
```

Kịch bản 5: Thay đổi tiêu đề mail

Sau khi dùng máy windows scanning để host bằng lệnh

```
Nmap -sS <ip_host>
```

```

Command Prompt
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>nmap -sS 192.168.202.156
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-15 22:20 SE Asia Standard Time
Nmap scan report for 192.168.202.156
Host is up (0.0034s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
MAC Address: 00:0C:29:34:FF:66 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.97 seconds

C:\Users\ACER>

```

Kịch bản 5: Quét cổng host sử dụng nmap

## Kết quả

Để có thể nhận mail ngay bây giờ ta sẽ thực hiện lệnh

```

quantc@quantc:~$ sudo logwatch --output mail --mailto tocongquan315@gmail.com --detail High
quantc@quantc:~$ 

```

Kịch bản 5: Gửi email ngay lập tức

The screenshot shows an email in the Gmail inbox with the following details:

- From:** warning for scanning (warning for scanning@localhost)
- To:** tocongquan315@gmail.com
- Subject:** Hộp thư đến
- Content:**

```

#####
Logwatch 7.5.6 (07/23/21) #####
Processing Initiated: Fri Nov 15 22:21:52 2024
Date Range Processed: yesterday
(2024-Nov-14 )
Period is day
Detail Level of Output: 10
Type of Output/Format: mail / text
Logfiles for Host: quantc
#####
----- Cron Begin -----
Commands Run:
User root:
cd / && run-parts --report /etc/cron.hourly: 1 Time(s)
[ -x /usr/lib/php/sessionclean ] && if [ ! -d /run/systemd/system ]; then /usr/lib/php/sessionclean; fi: 2 Time(s)
[ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi: 1 Time(s)
----- Cron End -----
----- dpkg status changes Begin -----

```

```

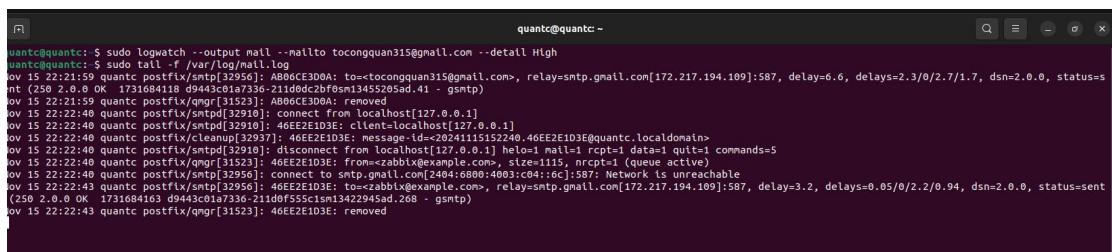
----- Kernel Begin -----

1 Time(s): ... bit width:      48
1 Time(s): ... event mask:    000100000000003f
1 Time(s): ... fixed-purpose events: 0
1 Time(s): ... generic registers: 6
1 Time(s): ... max period:    000000007fffff
1 Time(s): ... value mask:    0000ffffffffffff
1 Time(s): ... version:      1
1 Time(s): ... node #0, CPUs:   #1 #2 #3
1 Time(s): ..TIMER: vector=0x30 apic1=0 pin1=2 apic2=-1 pin2=-1
1 Time(s): /init
1 Time(s): ACPI: 1 ACPI AML tables successfully acquired and loaded
1 Time(s): ACPI: AC: AC Adapter [ACAD] (on-line)
1 Time(s): ACPI: APIC 0x00000000BFEDC872 000742 (v01 PTLTD ?APIC 06040000 LTP 00000000)
1 Time(s): ACPI: Added _OSI(3.0 _SCP Extensions)
1 Time(s): ACPI: Added _OSI(Module Device)
1 Time(s): ACPI: Added _OSI(Processor Aggregator Device)
1 Time(s): ACPI: Added _OSI(Processor Device)
1 Time(s): ACPI: BOOT 0x00000000BFEDCFB4 000028 (v01 PTLTD $SBFTBL$ 06040000 LTP 00000001)
1 Time(s): ACPI: Core revision 20230628
1 Time(s): ACPI: DSDT 0x00000000BFEDD001 021E72 (v01 PTLTD Custom 06040000 MSFT 03000001)
1 Time(s): ACPI: Early table checksum verification disabled
1 Time(s): ACPI: Enabled 4 GPEs in block 00 to 0F
1 Time(s): ACPI: FACP 0x00000000BFEEFEE73 0000F4 (v04 INTEL 440BX 06040000 PTL 000F4240)

```

Kịch bản 5: Xem email

### Xem Log của mail



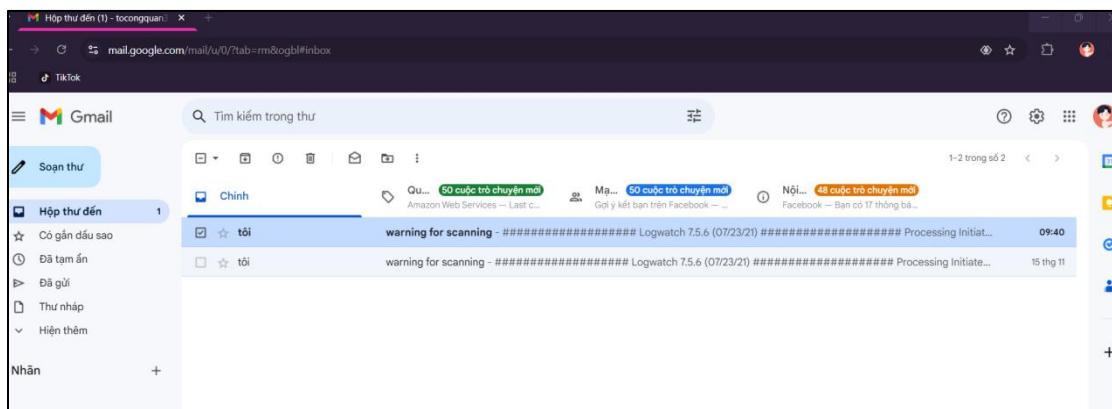
```

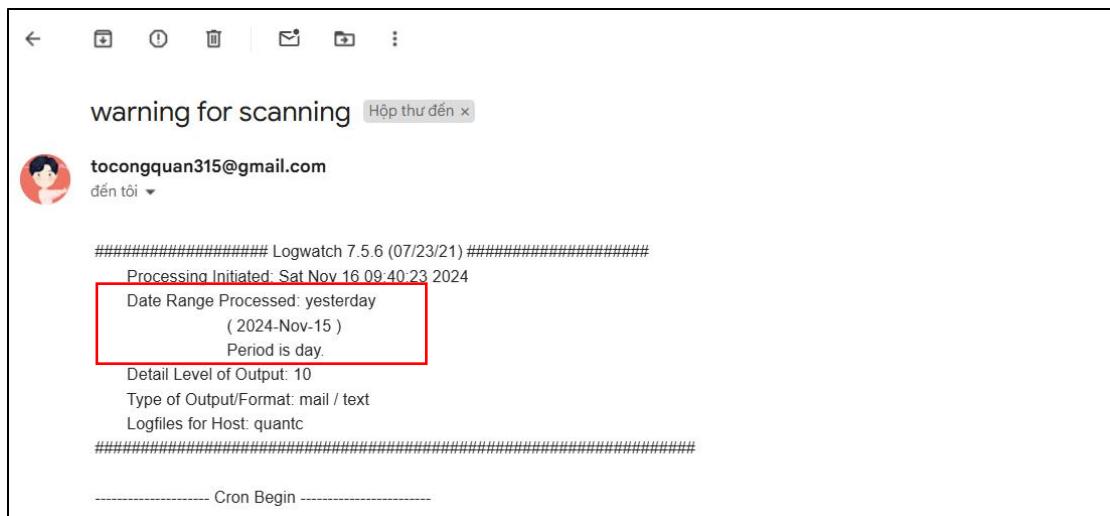
quantc@quantc: ~
quantc@quantc: $ sudo logwatch --output mail -mailto tocongquan315@gmail.com --detail High
quantc@quantc: $ sudo tail -f /var/log/mail.log
... (Log entries from logwatch output)

```

Kịch bản 5: Xem log của mail

Sau một ngày, ta sẽ nhận một email ghi log của nó ngày hôm qua





Kịch bản 5: Email được gửi sau một ngày

Link demo: [here](#)

## Kịch bản 6: Từ chối mọi kết nối chỉ giữ lại ping

Thêm các quy tắc sau:

- Cho phép gói ICMP loại echo-request (ping request) từ bên ngoài đến máy:

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

- Cho phép gói ICMP loại echo-reply (ping reply) từ máy đi ra ngoài:

```
sudo iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

- Chính sách mặc định cho INPUT và OUTPUT là DROP:

```
sudo iptables -P INPUT DROP
```

```
sudo iptables -P OUTPUT DROP
```

```

Dec 4 21:04
thaont@ubuntu: ~/NT131
thaont@ubuntu:~/NT131$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
thaont@ubuntu:~/NT131$ sudo iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
thaont@ubuntu:~/NT131$ sudo iptables -P OUTPUT DROP
thaont@ubuntu:~/NT131$ sudo iptables -P INPUT DROP
thaont@ubuntu:~/NT131$ sudo iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination
          0      0 ACCEPT    icmp  --  *       *      0.0.0.0/0            0.0.0.0/0           icmp type 8
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination
Chain OUTPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination
          0      0 ACCEPT    icmp  --  *       *      0.0.0.0/0            0.0.0.0/0           icmp type 0
thaont@ubuntu:~/NT131$ 
```

Kịch bản 6: Thêm các quy tắc

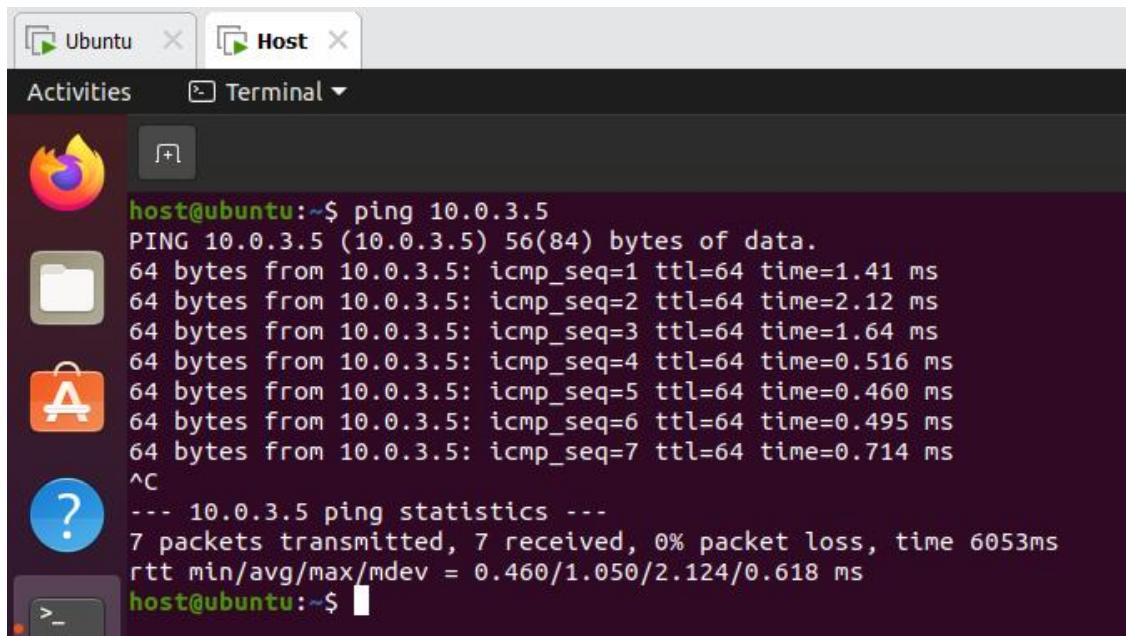
Kết quả hoạt động:

Đối với ping:

Các máy có thể nhận và trả lời ping với nhau nhờ quy tắc cho phép echo-request và echo-reply

Kết quả: Các máy sẽ thấy phản hồi khi ping

Ta thực hiện ping từ máy Host (10.0.3.8) đến máy Ubuntu (10.0.3.5) để kiểm tra các quy tắc:

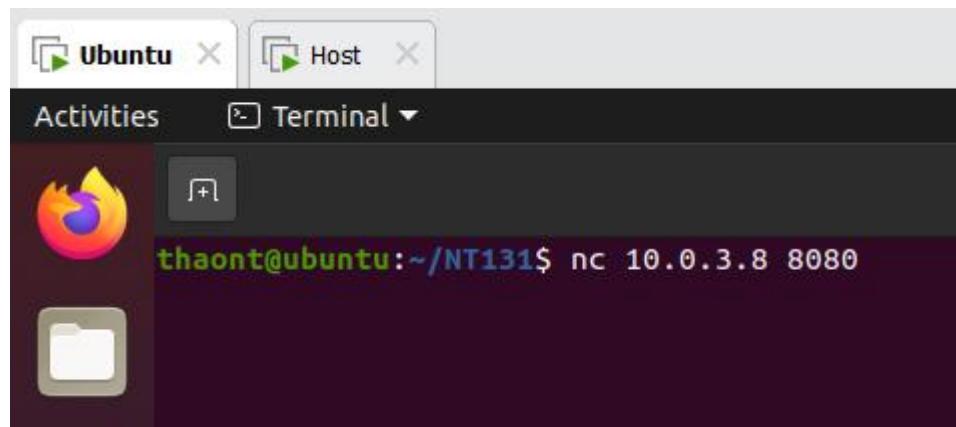


```
host@ubuntu:~$ ping 10.0.3.5
PING 10.0.3.5 (10.0.3.5) 56(84) bytes of data.
64 bytes from 10.0.3.5: icmp_seq=1 ttl=64 time=1.41 ms
64 bytes from 10.0.3.5: icmp_seq=2 ttl=64 time=2.12 ms
64 bytes from 10.0.3.5: icmp_seq=3 ttl=64 time=1.64 ms
64 bytes from 10.0.3.5: icmp_seq=4 ttl=64 time=0.516 ms
64 bytes from 10.0.3.5: icmp_seq=5 ttl=64 time=0.460 ms
64 bytes from 10.0.3.5: icmp_seq=6 ttl=64 time=0.495 ms
64 bytes from 10.0.3.5: icmp_seq=7 ttl=64 time=0.714 ms
^C
--- 10.0.3.5 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6053ms
rtt min/avg/max/mdev = 0.460/1.050/2.124/0.618 ms
host@ubuntu:~$
```

Kịch bản 6: Kết quả ping khi các quy tắc được áp dụng

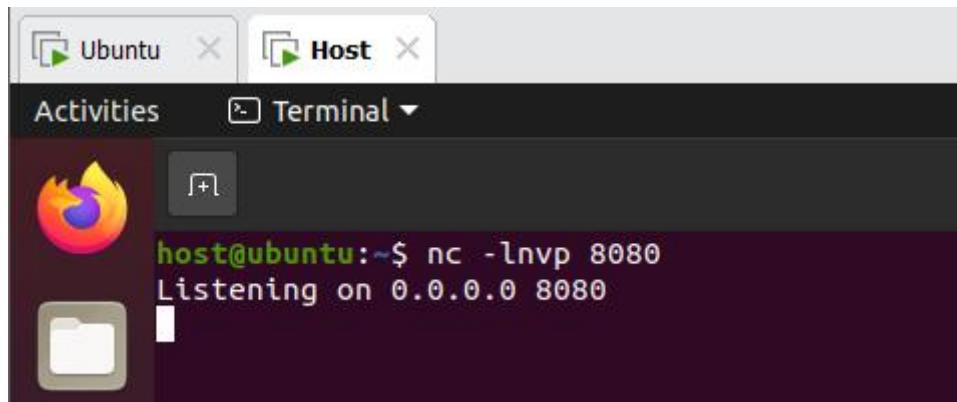
Đối với các kết nối khác:

- Không thể gửi gói tin ra ngoài: Chính sách OUTPUT DROP sẽ chặn tất cả gói tin không phải ICMP echo-reply
- Không nhận bất kỳ gói tin nào khác: Chính sách INPUT DROP sẽ chặn mọi gói tin không phải ICMP echo-request
- Kết quả: Kết nối TCP hoặc UDP đều bị chặn hoàn toàn
- Ta sử dụng netcat để thử kết nối từ máy Ubuntu (10.0.3.5) đến máy Host (10.0.3.8):



```
thaont@ubuntu:~/NT131$ nc 10.0.3.8 8080
```

Kịch bản 6: Sử dụng netcat để kết nối khi các quy tắc được áp dụng



Kịch bản 6: Kết quả là không thể kết nối được

Link demo: [here](#)

### Kịch bản 7: Chống DOS

Ta thiết lập các quy tắc trên máy Host (10.0.3.8)

#### a. Bảo vệ chống ICMP flood

```
sudo iptables -A INPUT -p icmp -m limit --limit 2/second --limit-burst 5 -j ACCEPT
```

```
sudo iptables -A INPUT -p icmp -j DROP
```

Giải thích:

- Chỉ chấp nhận tối đa 2 gói ICMP mỗi giây, với tối đa 5 gói burst
- Các gói ICMP vượt quá ngưỡng sẽ bị từ chối

#### b. Bảo vệ chống SYNflood

```
sudo iptables -A INPUT -p tcp --syn -m limit --limit 10/second --limit-burst 20 -j ACCEPT
```

```
sudo iptables -A INPUT -p tcp --syn -j DROP
```

Giải thích:

- Giới hạn lưu lượng SYN tối đa 10 gói mỗi giây với burst tối đa là 20 gói
- Tất cả lưu lượng SYN vượt quá ngưỡng trên sẽ bị từ chối

#### c. Bảo vệ chống UDPflood

```
sudo iptables -A INPUT -p udp -m limit --limit 10/second --limit-burst 20 -j ACCEPT
```

```
sudo iptables -A INPUT -p udp -j DROP
```

Giải thích:

- Chỉ chấp nhận tối đa 10 gói UDP mỗi giây, với tối đa 20 gói burst

- Các gói UDP vượt ngưỡng sẽ bị từ chối

```

host@ubuntu:~$ sudo iptables -A INPUT -p icmp -m limit --limit 2/second --limit-burst 5 -j ACCEPT
host@ubuntu:~$ sudo iptables -A INPUT -p icmp -j DROP
host@ubuntu:~$ sudo iptables -A INPUT -p tcp --syn -m limit --limit 10/second --limit-burst 20 -j ACCEPT
host@ubuntu:~$ sudo iptables -A INPUT -p tcp --syn -j DROP
host@ubuntu:~$ sudo iptables -A INPUT -p udp -m limit --limit 10/second --limit-burst 20 -j ACCEPT
host@ubuntu:~$ sudo iptables -A INPUT -p udp -j DROP
host@ubuntu:~$ sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out      source          destination
  0   0 ACCEPT      icmp  -- *       *       0.0.0.0/0        0.0.0.0/0      limit: avg 2/sec burst 5
  0   0 DROP        icmp  -- *       *       0.0.0.0/0        0.0.0.0/0
  0   0 ACCEPT      tcp   -- *       *       0.0.0.0/0        0.0.0.0/0      limit: avg 10/sec burst 20
  0   0 DROP        tcp   -- *       *       0.0.0.0/0        0.0.0.0/0      tcp flags:0x17/0x02
  0   0 ACCEPT      udp   -- *       *       0.0.0.0/0        0.0.0.0/0      limit: avg 10/sec burst 20
  0   0 DROP        udp   -- *       *       0.0.0.0/0        0.0.0.0/0      limit: avg 10/sec burst 20

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out      source          destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out      source          destination

```

Kịch bản 7: Thiết lập quy tắc

Thực hiện tấn công trên máy Ubuntu và kiểm tra các gói tin ở máy Host:

a. *Tấn công ICMP flood*

sudo ping -f 10.0.3.8

Giải thích: Gửi hàng loạt gói ICMP đến máy mục tiêu

b. *Tấn công SYN flood*

sudo hping3 -S --flood 10.0.3.8

Giải thích: Gửi liên tục gói SYN đến máy mục tiêu

c. *Tấn công UDP flood*

sudo hping3 --udp --flood 10.0.3.8

Giải thích: Gửi liên tục gói UDP đến máy mục tiêu

Link demo: [here](#)

**Kịch bản 8: Chỉ cho phép một địa chỉ cụ thể ssh vào host**

Ip máy thật windows

```

Ethernet adapter VMware Network Adapter VMnet8:

  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::6417:589c:fa87:3475%12
  IPv4 Address . . . . . : 192.168.17.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

Wireless LAN adapter Wi-Fi:

```

Kịch bản 8: Địa chỉ ip máy thật windows

Ip máy ảo windows

```
\Users\CsoL>ipconfig  
Windows IP Configuration  
  
Ethernet adapter Ethernet0:  
  
  Connection-specific DNS Suffix . : localdomain  
  Link-local IPv6 Address . . . . . : fe80::9277:58f2:3ed8:f60c%6  
  IPv4 Address. . . . . : 192.168.17.129  
  Subnet Mask . . . . . : 255.255.255.0  
  Default Gateway . . . . . : 192.168.17.2  
  
\Users\CsoL>
```

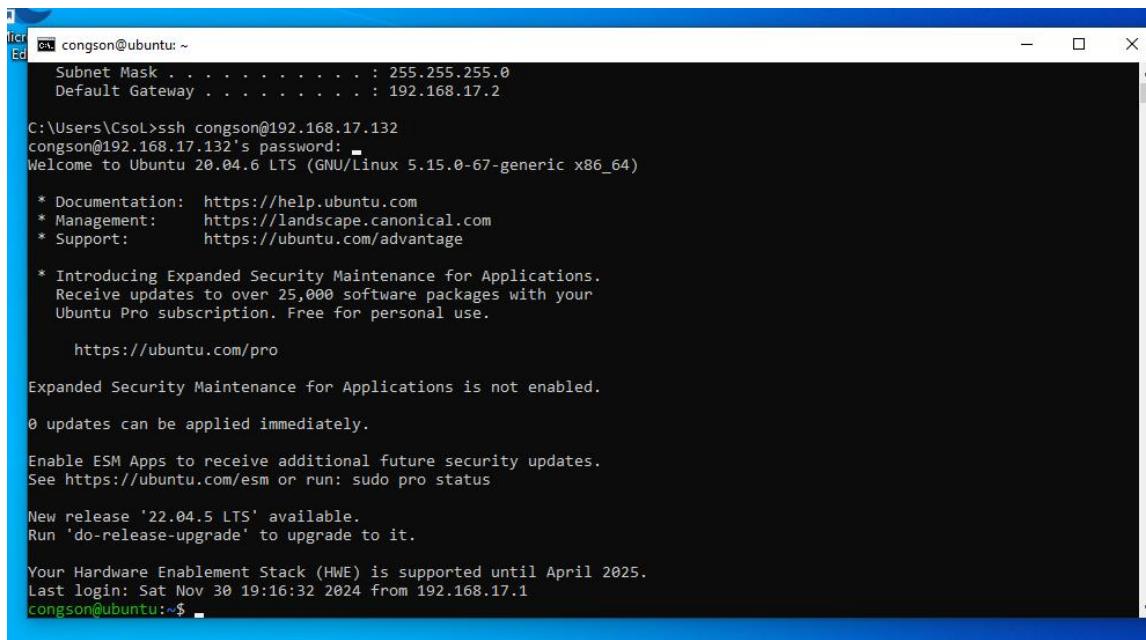
Kịch bản 8: Địa chỉ ip máy ảo windows

Ip máy host ubuntu

```
congson@ubuntu:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group  
    qlen 1000  
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
        inet 127.0.0.1/8 scope host lo  
            valid_lft forever preferred_lft forever  
        inet6 ::1/128 scope host  
            valid_lft forever preferred_lft forever  
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state  
    up default qlen 1000  
        link/ether 00:0c:29:ba:88:4e brd ff:ff:ff:ff:ff:ff  
        altname enp2s1  
        inet 192.168.17.132/24 brd 192.168.17.255 scope global dynamic nopref-  
    e ens33  
            valid_lft 1698sec preferred_lft 1698sec  
            inet6 fe80::7764:c839:ad09:163a/64 scope link noprefixroute  
                valid_lft forever preferred_lft forever  
congson@ubuntu:~$
```

Kịch bản 8: Địa chỉ ip host

Trước khi thiết lập quy tắc trên host ubuntu, ta thử ssh từ máy ảo windows đến host.



```
Administrator: ~$ ssh congson@192.168.17.132
congson@192.168.17.132's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-67-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

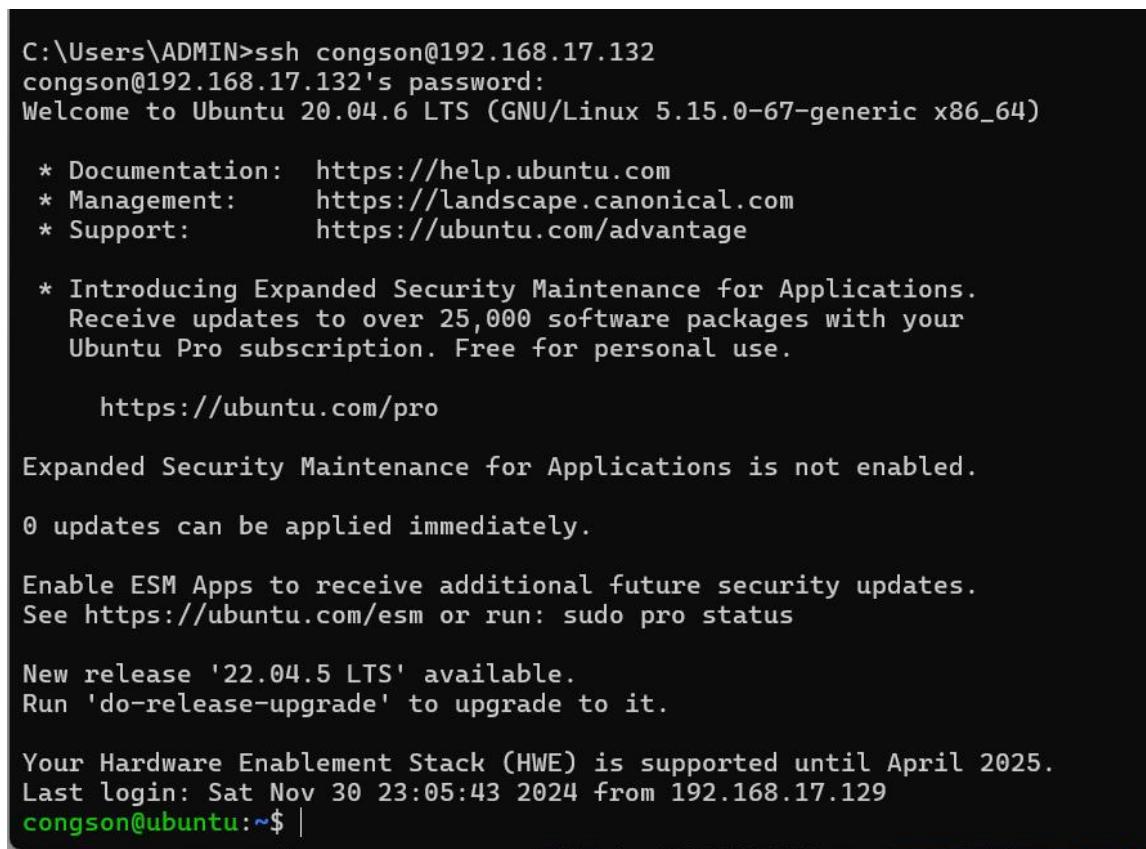
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Nov 30 19:16:32 2024 from 192.168.17.1
congson@ubuntu:~$
```

Kịch bản 8: SSH từ máy ảo windows đến host

Tiếp đó là SSH từ máy thật windows đến máy host ubuntu



```
C:\Users\ADMIN>ssh congson@192.168.17.132
congson@192.168.17.132's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-67-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Nov 30 23:05:43 2024 from 192.168.17.129
congson@ubuntu:~$ |
```

Kịch bản 8: SSH từ máy thật windows đến host

Tiến hành chạy kịch bản, kiểm tra các quy tắc trên máy host

```
valid_lft forever preferred_lft forever
congson@ubuntu:~$ sudo iptables -L
[sudo] password for congson:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
congson@ubuntu:~$
```

Kịch bản 8: Kiểm tra các quy tắc

Tiến hành cài đặt lệnh để chặn ssh từ máy ảo windows với IP là: 192.168.17.129

```
Try 'iptables -h' or 'iptables --help' for more information.
congson@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 22 -s 192.168.17.129 -j
DROP
congson@ubuntu:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP      tcp   --  192.168.17.129      anywhere            tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
congson@ubuntu:~$ S
```

Kịch bản 8: Thiết lập quy tắc chặn ip

Tiến hành SSH từ máy ảo windows đến máy host một lần nữa

```
Command Prompt
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\CsoL>ssh congson@192.168.17.132
ssh: connect to host 192.168.17.132 port 22: Connection timed out

C:\Users\CsoL>
```

Kịch bản 8: SSH từ máy ảo windows đến host lần nữa

Tiếp theo, sử dụng máy thật windows để ssh đến máy host

```

PS C:\Users\ADMIN> ssh congson@192.168.17.132
congson@192.168.17.132's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-67-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Nov 30 23:06:46 2024 from 192.168.17.1
congson@ubuntu:~$ |

```

Kịch bản: SSH từ máy thật windows đến host

Link demo: [here](#)

### Kịch bản 9: Chặn kết nối HTTP

Địa chỉ ip của máy chủ và đã cài đặt apache2 có cổng port là 80

```

congson@ubuntu:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-12-08 00:48:42 PST; 9min ago
     Docs: https://httpd.apache.org/docs/2.4/
          >Main PID: 8533 (apache2)
             Tasks: 55 (limit: 3368)
           Memory: 5.7M
          CGroup: /system.slice/apache2.service
                  ├─8533 /usr/sbin/apache2 -k start
                  ├─8534 /usr/sbin/apache2 -k start
                  ├─8535 /usr/sbin/apache2 -k start
                  └─8536 /usr/sbin/apache2 -k start

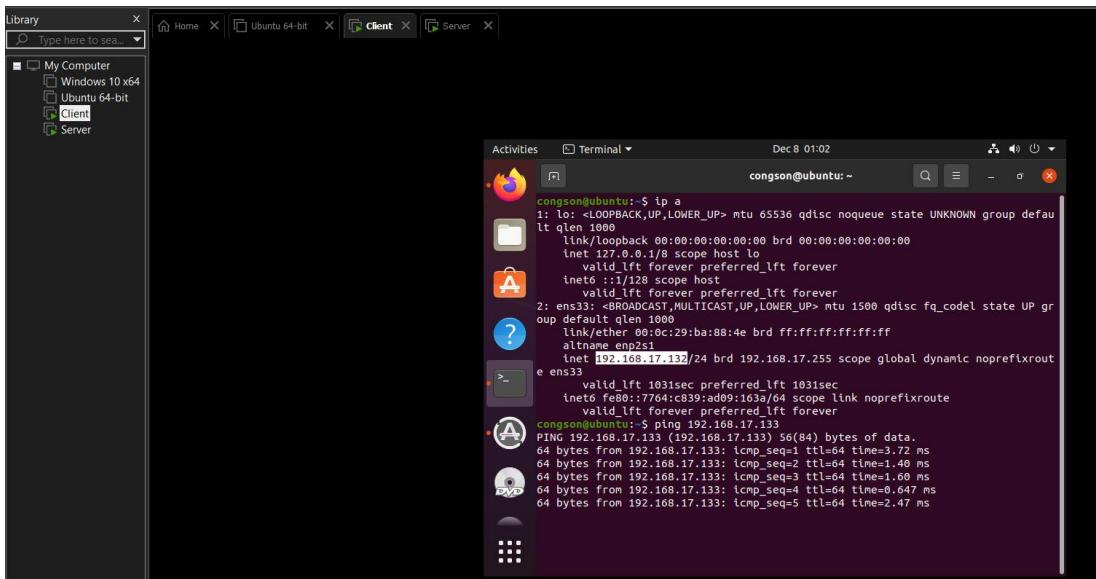
Dec 08 00:48:42 ubuntu systemd[1]: Starting The Apache HTTP Server...
Dec 08 00:48:42 ubuntu apachectl[8532]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, u
Dec 08 00:48:42 ubuntu systemd[1]: Started The Apache HTTP Server.

congson@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:fcc:aa brd ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.17.13/24 brd 192.168.17.255 scope global dynamic noprefixroute ens3
        valid_lft 1648sec preferred_lft 1648sec
    inet6 fe80::24d0:d5d4:ca8a:4934/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
congson@ubuntu:~$ ss

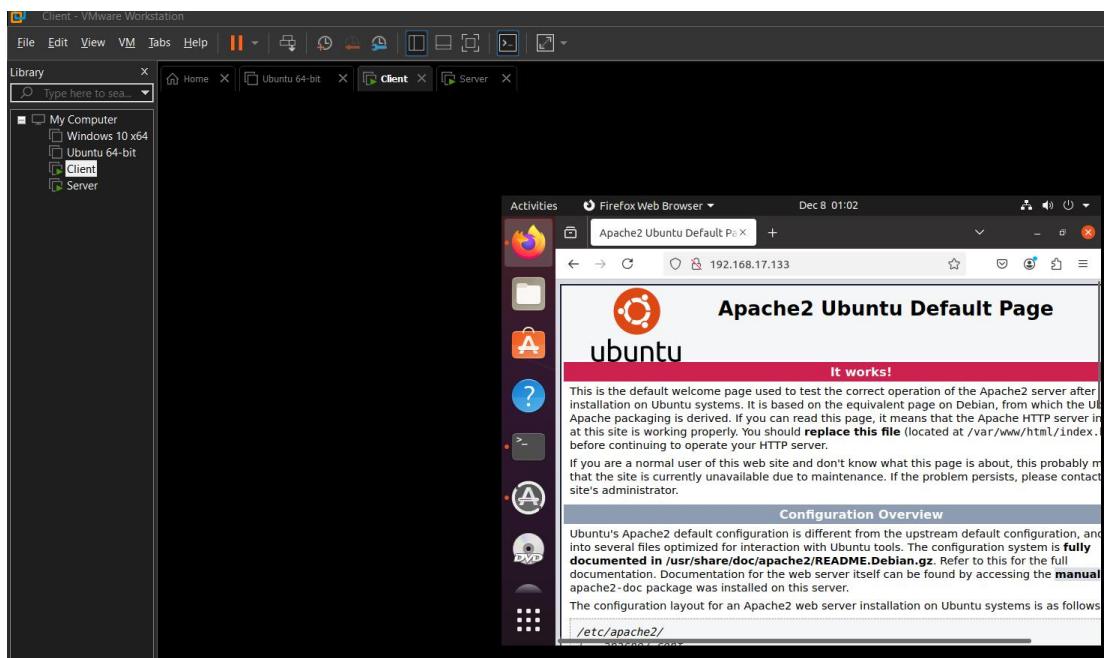
```

Kịch bản 9: Kiểm tra địa chỉ ip máy host

Địa chỉ ip của máy client và kết nối tới server apache2 của máy chủ thành công



Kịch bản 9: Kiểm tra địa chỉ IP máy client



Kịch bản 9: Truy cập website Apache trên client

Bây giờ, thiết lập quy tắc chặn client trên host

```

congson@ubuntu:~$ sudo systemctl status apache2
● apache2.service - Apache2
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-12-08 00:48:42 PST; 17min ago
     Docs: https://httpd.apache.org/docs/2.4/
          >Main PID: 8533 (apache2)
             Tasks: 5 (llimit: 3368)
            Memory: 8.40M
           CGroup: /system.slice/apache2.service
                   ├─8533 /usr/sbin/apache2 -k start
                   ├─8534 /usr/sbin/apache2 -k start
                   ├─8535 /usr/sbin/apache2 -k start
                   └─8536 /usr/sbin/apache2 -k start

Dec 08 00:48:42 ubuntu systemd[1]: Starting The Apache HTTP Server...
Dec 08 00:48:42 ubuntu apachectl[8532]: AH00558: Apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive
Lines 1-15/15 (END)

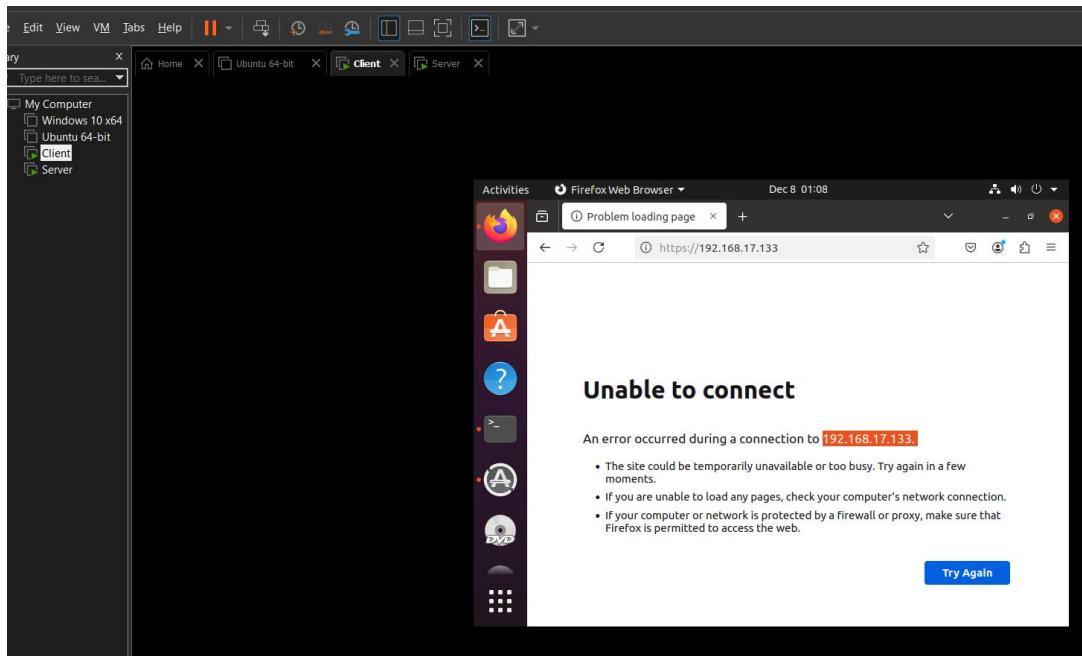
congson@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 80 -s 192.168.17.132 -j REJECT
congson@ubuntu:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 7 packets, 1031 bytes)
pkts bytes target  prot opt in  out source      destination
  0   0 REJECT  tcp  --  any  any  192.168.17.132 anywhere    tcp dpt:http reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target  prot opt in  out source      destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target  prot opt in  out source      destination
congson@ubuntu:~$
```

Kịch bản 9: Thiết lập quy tắc

Kết quả sau khi chặn



Kịch bản 9: Kết quả sau khi chặn

Link demo: [here](#)

## IV. KẾT LUẬN

Trong báo cáo này, nhóm đã nghiên cứu và tìm hiểu về khái niệm, vai trò, và các thành phần cơ bản của Iptables và Netfilter, bao gồm các loại bảng, chain, và target của Iptables cũng như các hook quan trọng của Netfilter. Việc ứng dụng các kịch bản thực tế cho thấy hiệu quả của Netfilter và Iptables trong việc thiết lập các quy tắc bảo mật, quản lý lưu lượng mạng, và ngăn chặn các tấn công mạng phổ biến như DoS và DDoS.

Việc nắm vững mô hình TCP/IP và cách thức hoạt động của Netfilter giúp nâng cao khả năng bảo mật và hiệu suất mạng của hệ thống. Từ những kiến thức này, nhóm đã triển khai thành công nhiều kịch bản thực tế, đóng góp vào việc nâng cao an ninh mạng cho hệ thống Linux.

Trong tương lai, có thể mở rộng nghiên cứu và ứng dụng các công nghệ mới như nftables hoặc tiến sâu hơn vào các khái niệm mạng phức tạp hơn như SDN (Software Defined Networking) để tối ưu hóa và bảo vệ mạng.

# BẢNG PHÂN CÔNG

| STT | Họ và tên         | MSSV     | Phân công                                        |
|-----|-------------------|----------|--------------------------------------------------|
| 1   | Tô Công Quân      | 22521190 | Netfilter: Kịch bản 1<br>Iptables: Kịch bản 4,5  |
| 2   | Nguyễn Thành Thạo | 22521371 | Netfilter: kịch bản 2<br>Iptables: kịch bản 6, 7 |
| 3   | Đào Công Sơn      | 22521249 | Netfilter: Kịch bản 3<br>Iptables: kịch bản 8, 9 |

# TÀI LIỆU THAM KHẢO

- [1] J. Ellingwood, "A Deep Dive into Iptables and Netfilter Architecture," 2 Nov 2022. [Online]. Available: <https://www.digitalocean.com/community/tutorials/a-deep-dive-into-iptables-and-netfilter-architecture>.
- [2] M. K. Thummaluru, "Introduction to Netfilter," 4 April 2024. [Online]. Available: <https://blogs.oracle.com/linux/post/introduction-to-netfilter>.
- [3] rostectg, "NT131," June 2024. [Online]. Available: <https://github.com/rostectg/NT131>.
- [4] Nhom5, "Lab03\_Nhom5," 14 Oct 2024. [Online]. Available: <https://drive.google.com/file/d/1OVxApy2NhFRV0ww1GVsJDSufwWtQCxSu/view>.