

TRƯỜNG ĐẠI HỌC KHOA HỌC  
KHOA CÔNG NGHỆ THÔNG TIN



**AN NINH MẠNG**

**Đề tài:**

**TÌM HIỂU TẦN CÔNG DOS/DDOS VÀ CÁCH THỨC  
PHÒNG CHỐNG**

**Tên học phần: An ninh mạng – Nhóm 2**

**Mã lớp học phần: 2025-2026.1.TIN3163.002**

**Giảng viên hướng dẫn: Võ Việt Dũng**

**Sinh viên thực hiện: 23T1020111 - Hoàng Minh Đức**

**23T1020191 - Hồ Văn Huấn**

**23T1020248 - Bùi Nguyễn Nhật Huy**

**23T1020582 - Phạm Quang Tuân**

**21T1020361 - Nguyễn Công Hiếu**

*Huế, ngày 05 tháng 1 năm 2026*

## MỤC LỤC

<b>CHƯƠNG 1 - KHẢO SÁT HỆ THỐNG .....</b>	6
1.1 Lý do chọn đề tài .....	6
1.2 Mục tiêu nghiên cứu .....	6
1.3 Đối tượng và phạm vi nghiên cứu .....	6
1.4 Phương pháp nghiên cứu .....	6
1.5 Bộ cục của bài tiểu luận .....	7
<b>CHƯƠNG 2 - TỔNG QUAN VỀ TẤN CÔNG DOS/DDOS.....</b>	7
2.1 Khái niệm tấn công DoS .....	7
2.2 Khái niệm tấn công DDoS .....	8
2.3 Phân biệt DoS và DDoS.....	8
2.4 Nguyên lý hoạt động chung của tấn công DoS/DDoS .....	9
2.5 Tác động của tấn công DoS/DDoS đối với hệ thống mạng.....	10
<b>CHƯƠNG 3 - CÁC KỸ THUẬT TẤN CÔNG DOS/DDOS PHỔ BIẾN .....</b>	11
3.1 Ping of Death.....	11
3.1.1 Khái niệm .....	11
3.1.2 Cơ chế hoạt động .....	11
3.1.3 Tính chất.....	12
3.1.4 Tình trạng hiện nay và biến thể .....	12
3.1.5 Cách phòng chống .....	13
3.2 Teardrop Attack .....	13
3.2.1 Khái niệm .....	13
3.2.2 Cơ chế hoạt động .....	14
3.2.3 Tính chất.....	15
3.2.4 Tình trạng hiện nay và biến thể .....	15
3.2.5 Cách phòng chống .....	15
3.3 TCP SYN Flood.....	16
3.3.1 Khái niệm .....	16

3.3.2 Cơ chế hoạt động .....	16
3.3.3 Tính chất.....	18
3.3.4 Tình trạng hiện nay và biến thể .....	19
3.3.5 Cách phòng chống: .....	19
3.4 DNS Amplification Attack .....	20
3.4.1 Khái niệm .....	20
3.4.2 Cơ chế hoạt động .....	20
3.4.3 Tính chất.....	22
3.4.4 Tình trạng hiện nay và biến thể .....	23
3.4.5 Cách phòng chống .....	23
3.5 Đặc điểm và xu hướng tấn công DoS/DDoS hiện nay .....	25
3.5.1 Đặc điểm chung của các cuộc tấn công hiện đại .....	25
3.5.2. Xu hướng chuyển dịch kỹ thuật .....	25
3.5.3. Quy mô và tần suất kỷ lục .....	26
3.5.4 Hình thức tổ chức và động cơ .....	26
3.5.5 Xu hướng các Vector tấn công cụ thể .....	26
3.5.6 Tổng kết: So sánh kỹ thuật cổ điển và hiện đại .....	27
<b>CHƯƠNG 4 - CÁCH THỨC TẤN CÔNG DOS/DDOS VÀ MÔ HÌNH BOTNET .....</b>	<b>27</b>
4.1 Cách thức tấn công DoS .....	27
4.1.1 Tấn công SYN Flood .....	27
4.1.2 Tấn công ICMP Flood (Ping Flood).....	28
4.1.3 Tấn công UDP Flood .....	29
4.1.4 Tấn công HTTP Flood .....	30
4.2 Cách thức tấn công DDoS.....	31
4.3 Mô hình botnet tấn công trong DDoS .....	31
4.3.1 Tổng quan.....	31
4.3.2 Các thành phần trong mô hình botnet.....	31
4.3.3 Quy trình hình thành botnet (Giai đoạn chuẩn bị) .....	31
4.3.4 Quy trình tấn công DDoS sử dụng botnet .....	32

4.3.5 Đặc điểm nguy hiểm của mô hình botnet .....	32
4.4 Nhận xét và đánh giá .....	33
<b>CHƯƠNG 5 - CÁC BIỆN PHÁP PHÒNG CHỐNG TẤN CÔNG DOS/DDOS .....</b>	<b>33</b>
5.1 Phòng chống ở mức mạng (Network Level).....	33
5.2 Phòng chống ở mức hệ điều hành (Operating System Level) .....	34
5.3 Phòng chống ở mức ứng dụng (Application Level).....	36
5.4 Phòng chống bằng dịch vụ và giải pháp chuyên dụng .....	36
5.5. So sánh các biện pháp phòng chống.....	37
<b>CHƯƠNG 6 - THỰC HÀNH, MÔ PHỎNG VÀ ĐÁNH GIÁ .....</b>	<b>38</b>
6.1 Mục tiêu thí nghiệm.....	38
6.2 Mô hình và môi trường thí nghiệm .....	38
6.2.1 Mô hình mạng.....	38
6.2.2 Công cụ sử dụng .....	39
6.3 Chuẩn bị môi trường.....	40
6.3.1 Kiểm tra kết nối mạng .....	40
6.3.2 Cài đặt và kiểm tra Nginx (Ubuntu) .....	41
6.3.3 Đo baseline (trước tấn công) .....	42
6.4 Tiến hành tấn công SYN FLOOD .....	43
6.4.1 Thực hiện tấn công .....	44
6.4.2 Kết thúc tấn công .....	45
6.4.3 Nhận xét .....	45
6.5 Phòng thủ SYN FLOOD (Layer 4) .....	45
6.5.1 Bật SYN Cookies (Ubuntu) .....	46
6.5.2 Thêm rule iptables giới hạn SYN (Ubuntu).....	46
6.5.3 Chuẩn bị giám sát .....	46
6.5.4 Chạy lại SYN Flood (Kali) .....	47
6.5.5 Kết thúc tấn công .....	48
6.5.6 Nhận xét .....	48
6.7 Tấn công HTTP FLOOD .....	49

6.7.1. Chuẩn bị giám sát .....	49
6.7.2 Chuẩn bị tấn công (Kali).....	50
6.7.3 Chạy HTTP Flood.....	51
6.7.4 Kết thúc tấn công.....	51
6.7.5 Nhận xét .....	51
6.8 Phòng thủ HTTP FLOOD (Layer 7).....	52
6.8.1 Mở file cấu hình Nginx (Ubuntu).....	52
6.8.2 Thêm zone giới hạn request (HTTP context).....	53
6.8.3 Áp dụng limit cho server (server/location) .....	53
6.8.4 Kiểm tra & reload Nginx (Không restart).....	54
6.8.5 Chuẩn bị giám sát .....	54
6.8.6 Chạy lại HTTP Flood (Kali).....	55
6.8.7 Dừng.....	55
6.8.8 Nhận xét .....	55
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>57</b>

## LỜI MỞ ĐẦU

Bài toán tìm hiểu và phân tích các hình thức tấn công DoS/DDoS trong mạng máy tính được đặt ra nhằm làm rõ bản chất, nguyên lý hoạt động và mức độ ảnh hưởng của các cuộc tấn công từ chối dịch vụ đối với hệ thống mạng và dịch vụ trực tuyến hiện nay. Trong bối cảnh Internet phát triển mạnh mẽ, các hệ thống thông tin ngày càng phụ thuộc vào hạ tầng mạng, việc đảm bảo tính sẵn sàng và ổn định của dịch vụ trở thành yêu cầu quan trọng đối với cá nhân, doanh nghiệp và tổ chức

Tấn công DoS/DDoS là một trong những mối đe dọa phổ biến và nguy hiểm, có khả năng làm gián đoạn hoặc tê liệt hoàn toàn hoạt động của hệ thống bằng cách khai thác và làm quá tải tài nguyên mạng, máy chủ hoặc ứng dụng. Các cuộc tấn công này không chỉ gây thiệt hại về kinh tế mà còn ảnh hưởng đến uy tín, độ tin cậy và an toàn thông tin của tổ chức bị tấn công

Đề tài tập trung nghiên cứu các khái niệm cơ bản về DoS và DDoS, phân tích sự khác biệt giữa hai hình thức tấn công, các phương thức tấn công phổ biến cũng như quy trình thực hiện của một cuộc tấn công từ chối dịch vụ. Thông qua đó, đề tài giúp người học hiểu rõ hơn về cách thức mà các cuộc tấn công này diễn ra trong môi trường mạng thực tế

Bên cạnh đó, đề tài cũng đi sâu vào việc tìm hiểu các giải pháp và biện pháp phòng chống tấn công DoS/DDoS, bao gồm các phương pháp kỹ thuật, cơ chế giám sát và bảo vệ hệ thống mạng. Việc nghiên cứu các biện pháp phòng chống góp phần nâng cao khả năng phát hiện sớm, giảm thiểu rủi ro và hạn chế thiệt hại do các cuộc tấn công gây ra

Thông qua việc thực hiện đề tài, người viết mong muốn nâng cao nhận thức về an toàn và bảo mật mạng, đồng thời cung cấp cái nhìn tổng quan và hệ thống về tấn công DoS/DDoS cũng như tầm quan trọng của việc xây dựng các giải pháp phòng chống hiệu quả trong môi trường công nghệ thông tin hiện nay

# CHƯƠNG 1 - KHẢO SÁT HỆ THỐNG

## 1.1 Lý do chọn đề tài

Trong kỷ nguyên công nghệ số hiện nay, Internet đã trở thành nền tảng không thể thiếu đối với mọi hoạt động kinh tế, xã hội và giáo dục. Tuy nhiên, sự phát triển này cũng đi kèm với những nguy cơ mất an toàn thông tin ngày càng gia tăng. Một trong những hình thức tấn công mạng kinh điển nhưng vẫn gây ra hậu quả nghiêm trọng nhất là Tấn công từ chối dịch vụ (Denial of Service - DoS) và Tấn công từ chối dịch vụ phân tán (Distributed Denial of Service - DDoS).

Các cuộc tấn công này không nhằm mục đích đánh cắp dữ liệu, mà nhằm làm tê liệt hệ thống, khiến người dùng hợp pháp không thể truy cập dịch vụ, gây thiệt hại to lớn về kinh tế và uy tín cho các tổ chức. Chính vì tính chất nguy hiểm và phổ biến của chúng, việc tìm hiểu sâu về cơ chế hoạt động cũng như các biện pháp phòng chống DoS/DDoS là vô cùng cấp thiết đối với sinh viên ngành An toàn thông tin. Đó là lý do em chọn đề tài "Tìm hiểu tấn công DoS/DDoS trong mạng và cách thức phòng chống" để nghiên cứu.

## 1.2 Mục tiêu nghiên cứu

Mục tiêu của đề tài bao gồm:

- Trình bày các khái niệm cơ bản về tấn công DoS và DDoS trong mạng máy tính
- Phân tích nguyên lý hoạt động và các kỹ thuật tấn công DoS/DDoS phổ biến
- Làm rõ sự khác biệt giữa tấn công DoS và tấn công DDoS
- Đánh giá tác động của các cuộc tấn công DoS/DDoS đối với hệ thống mạng và dịch vụ trực tuyến
- Nghiên cứu và tổng hợp các biện pháp phòng chống tấn công DoS/DDoS ở nhiều mức độ khác nhau
- Góp phần nâng cao nhận thức về an toàn và bảo mật mạng trong môi trường công nghệ thông tin hiện nay

## 1.3 Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Các hình thức tấn công DoS/DDoS trong mạng máy tính và các giải pháp phòng chống tương ứng
- Phạm vi nghiên cứu: Đề tài tập trung nghiên cứu DoS/DDoS ở mức độ lý thuyết và mô phỏng, bao gồm khái niệm, nguyên lý hoạt động, các kỹ thuật tấn công phổ biến và các biện pháp phòng chống thường được áp dụng. Đề tài không đi sâu vào việc triển khai hệ thống phòng thủ quy mô lớn trong môi trường doanh nghiệp thực tế.

## 1.4 Phương pháp nghiên cứu

Để thực hiện đề tài, các phương pháp nghiên cứu sau được sử dụng:

- Thu thập, nghiên cứu và tổng hợp tài liệu từ sách, giáo trình, bài báo khoa học và các nguồn tham khảo uy tín liên quan đến an ninh mạng và DoS/DDoS
- Phân tích và so sánh các hình thức tấn công DoS và DDoS
- Mô phỏng và đánh giá đặc điểm của một số kỹ thuật tấn công DoS/DDoS

- Tổng hợp và đánh giá các biện pháp phòng chống dựa trên lý thuyết và thực tiễn
- Trình bày kết quả nghiên cứu một cách hệ thống, logic và dễ hiểu

## 1.5 Bộ cục của bài tiểu luận

Nội dung được tổ chức thành 6 chương, cụ thể như sau:

Chương 1: Mở đầu – Giới thiệu lý do chọn đề tài, mục tiêu, đối tượng, phạm vi và phương pháp nghiên cứu

Chương 2: Tổng quan về tấn công DoS/DDoS – Trình bày các khái niệm, nguyên lý hoạt động và tác động của tấn công DoS/DDoS

Chương 3: Các kỹ thuật tấn công DoS/DDoS phổ biến – Phân tích các kỹ thuật tấn công điển hình và xu hướng hiện nay

Chương 4: Cách thức tấn công DoS/DDoS – Mô tả chi tiết quy trình và mô hình tấn công, đặc biệt là mô hình botnet

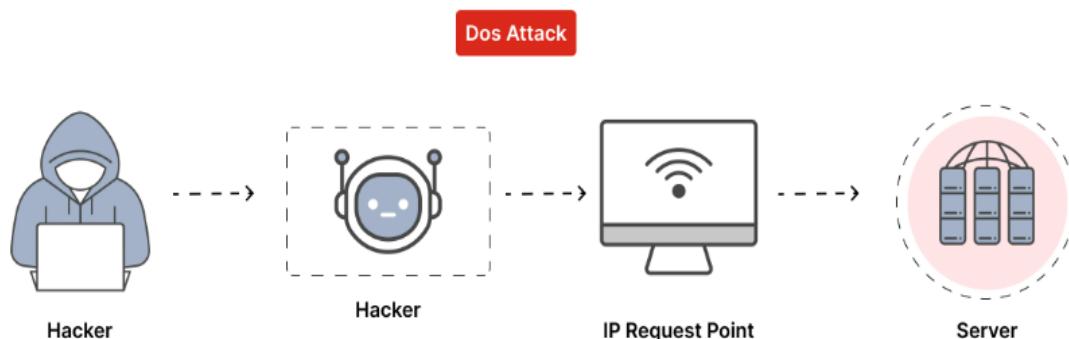
Chương 5: Các biện pháp phòng chống tấn công DoS/DDoS – Trình bày và so sánh các giải pháp phòng chống ở nhiều mức độ

Chương 6: Thực hành – mô phỏng và đánh giá – Mô phỏng tấn công và thử nghiệm một số biện pháp phòng chống

## CHƯƠNG 2 - TỔNG QUAN VỀ TẤN CÔNG DOS/DDOS

### 2.1 Khái niệm tấn công DoS

- DoS viết tắt của cụm từ Denial of Service, là một kiểu tấn công từ chối dịch vụ khi đó máy tính của bạn sẽ bị tấn công bởi lưu lượng truy cập từ hệ thống của hacker. DoS là một cuộc tấn công trực tuyến thường nhắm vào một trang web hoặc máy chủ điển hình. Bằng cách làm quá tải tài nguyên hệ thống, tốc độ hệ thống của máy tính sẽ bị chậm lại đáng kể
- Cuộc tấn công này có thể khiến máy tính của bạn ngừng hoạt động hoặc tắt đột ngột. Khi hiện tượng này xảy ra sẽ ảnh hưởng nghiêm trọng đến hệ thống của máy tính và buộc máy tính phải tắt nguồn [2.1]

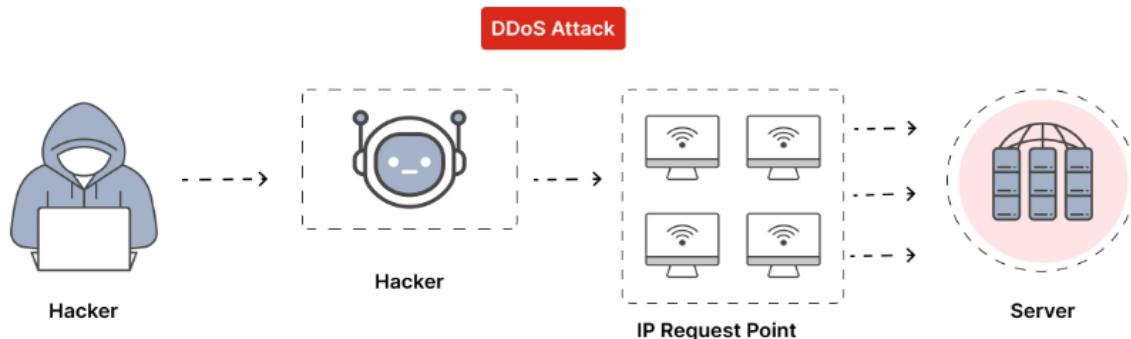


Hình 2.1 Biểu đồ mô tả công cụ DoS

## 2.2 Khái niệm tấn công DDoS

- DDoS viết tắt của cụm từ Distributed Denial of Service, có nghĩa là từ chối dịch vụ phân tán, máy tính của bạn bị tấn công với lưu lượng truy cập từ nhiều hệ thống khác nhau thông qua nhiều nơi khác nhau

- Trọng tâm của việc này là máy tính hoặc server chắc chắn sẽ bị đánh sập hoặc ngừng hoạt động, gián đoạn dịch vụ. Những kẻ tấn công sau khi có được quyền kiểm soát máy tính sẽ lợi dụng điều đó để gửi các dữ liệu xấu, các yêu cầu đến các thiết bị khác thông qua trang web hoặc địa chỉ email [2.1]



Hình 2.2 Biểu đồ mô tả DDoS

## 2.3 Phân biệt DoS và DDoS

Tấn công từ chối dịch vụ (DoS) và tấn công phân tán từ chối dịch vụ (DDoS) là những mối đe dọa ngày càng gia tăng trong thế giới công nghệ hiện nay. Những cuộc tấn công này có thể gây gián đoạn dịch vụ, thậm chí khiến hệ thống của doanh nghiệp bị tê liệt trong thời gian dài, gây thiệt hại tài chính và ảnh hưởng nghiêm trọng đến uy tín của công ty. Để đối phó với những thách thức này, việc hiểu rõ sự khác biệt giữa DoS và DDoS, cũng như các biện pháp bảo vệ, là rất quan trọng. Bảng dưới đây sẽ giúp ta so sánh và phân tích chi tiết giữa DoS và DDoS, từ đó đưa ra giải pháp bảo vệ hiệu quả cho doanh nghiệp [2.2]

Tiêu chí	DoS Attack	DDoS Attack
Nguồn tấn công	Một nguồn duy nhất (1 máy tính, 1 địa chỉ IP)	<b>DDoS (Distributed Denial of Service):</b> Nhiều máy (thường là botnet) đồng loạt tấn công, gây thiệt hại nghiêm trọng hơn.
Quy mô tấn công	Quy mô nhỏ, dễ dàng phát hiện	Quy mô lớn, khó phát hiện và ngăn chặn

<b>Phát hiện và ngăn chặn</b>	DỄ phát hiện, có thể chặn IP hoặc yêu cầu	Khó phát hiện, vì có nhiều nguồn tấn công
<b>Khả năng gây thiệt hại</b>	Tác động ngắn hạn, gián đoạn dịch vụ tạm thời	Tác động lâu dài, có thể làm tê liệt hệ thống
<b>Tính hiệu quả</b>	Ít hiệu quả, chỉ gián đoạn trong thời gian ngắn	Rất hiệu quả, làm tê liệt dịch vụ trong thời gian dài
<b>Ảnh hưởng đến doanh nghiệp</b>	Thường chỉ gián đoạn dịch vụ ngắn hạn	Thiệt hại tài chính lớn, mất uy tín lâu dài
<b>Giải pháp bảo vệ</b>	Firewall cơ bản, chặn địa chỉ IP tấn công	Anti-DDoS, giải pháp tường lửa phân tán, bảo vệ cloud Thiệt hại tài chính lớn, mất uy tín lâu dài

## 2.4 Nguyên lý hoạt động chung của tấn công DoS/DDoS

- Để hiểu rõ vì sao tấn công DDoS có thể khiến hệ thống sập chỉ trong thời gian ngắn, doanh nghiệp cần nắm được cơ chế hoạt động cơ bản của hình thức tấn công từ chối dịch vụ này. Dưới đây là một quy trình tấn công DDoS thông thường:

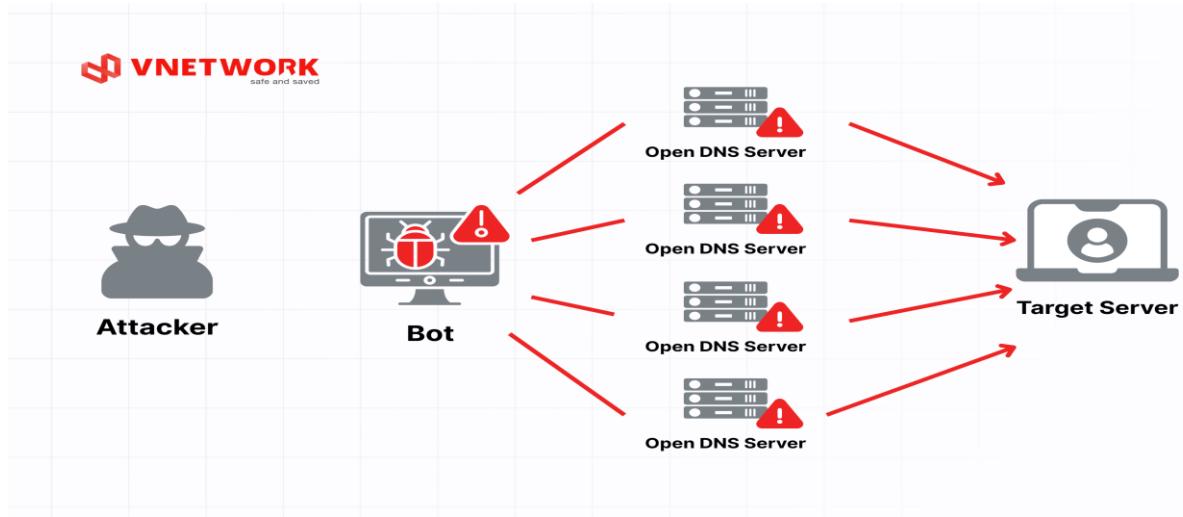
- Xây dựng botnet: Kẻ tấn công trước hết kiểm soát một số lượng lớn thiết bị kết nối Internet như máy tính, máy chủ hoặc thiết bị IoT thông qua mã độc, biến chúng thành các “bot độc” hoạt động dưới sự điều khiển từ xa. Đây là nền tảng để triển khai tấn công DDoS trên quy mô lớn

- Điều khiển tập trung: Thông qua hệ thống điều khiển trung tâm, kẻ tấn công ra lệnh cho toàn bộ botnet cùng lúc nhắm đến một địa chỉ IP hoặc hệ thống mục tiêu cụ thể. Nhờ tính phân tán, lưu lượng truy cập trong tấn công từ chối dịch vụ có thể đến từ nhiều quốc gia và nhiều dải IP khác nhau

- Gửi lưu lượng ồ ạt về mục tiêu: Mỗi bot sẽ liên tục gửi các yêu cầu kết nối hoặc gói dữ liệu về hệ thống bị tấn công, tạo ra lượng truy cập lớn bất thường. Khi số lượng bot đủ lớn, DDoS server phải tiếp nhận lưu lượng vượt xa khả năng xử lý thông thường

- Làm cạn kiệt tài nguyên hệ thống: Trong quá trình xử lý các yêu cầu không hợp lệ, tài nguyên của máy chủ như băng thông, CPU hoặc bộ đệm kết nối bị chiếm dụng nhanh chóng. Điều này khiến hệ thống không còn đủ khả năng phản hồi các yêu cầu hợp lệ từ người dùng thật

- Từ chối dịch vụ với người dùng hợp lệ: Khi tài nguyên bị quá tải, website hoặc ứng dụng sẽ trở nên chậm, lỗi hoặc hoàn toàn không thể truy cập. Kết quả cuối cùng của DDoS là dịch vụ bị gián đoạn, gây ảnh hưởng trực tiếp đến trải nghiệm người dùng và hoạt động kinh doanh [2.3]



Hình 2.3 Quy trình tấn công DDoS

## 2.5 Tác động của tấn công DoS/DDoS đối với hệ thống mạng

Tấn công từ chối dịch vụ (DoS) có thể gây ra những tác hại nghiêm trọng đối với hệ thống, dịch vụ và doanh nghiệp. Dưới đây là những tác động chính mà một cuộc tấn công DoS có thể gây ra:

- Gián đoạn hoạt động kinh doanh: Mục tiêu chính của DoS là làm gián đoạn các dịch vụ trực tuyến hoặc hệ thống mạng, khiến người dùng không thể truy cập vào các dịch vụ quan trọng. Điều này có thể dẫn đến mất doanh thu trong khi hệ thống không thể phục vụ khách hàng hoặc thực hiện các giao dịch
- Mất uy tín và niềm tin của khách hàng: Khi khách hàng không thể truy cập dịch vụ, họ sẽ cảm thấy thất vọng và mất niềm tin vào chất lượng dịch vụ của doanh nghiệp
- Tăng chi phí khắc phục: Sau khi bị tấn công DoS, doanh nghiệp cần phải chi tiêu một khoản lớn để phục hồi hệ thống, bao gồm việc cải thiện cơ sở hạ tầng, nâng cấp các biện pháp bảo mật và thuê nhân lực chuyên môn. Những khoản chi này có thể ảnh hưởng đến lợi nhuận và tài chính của công ty
- Tổn thất về dữ liệu và thông tin: Trong một số trường hợp, các cuộc tấn công DoS có thể tạo điều kiện cho các loại tấn công khác (như DDoS hoặc tấn công SQL Injection), dẫn đến mất mát dữ liệu quan trọng hoặc lộ thông tin nhạy cảm, ảnh hưởng đến sự bảo mật của tổ chức
- Hệ thống bị quá tải và hoạt động chậm chạp: Khi hệ thống bị tấn công DoS, các tài nguyên như băng thông mạng, bộ nhớ và CPU có thể bị quá tải. Điều này không chỉ

làm hệ thống không hoạt động hiệu quả mà còn có thể gây ra trễ dịch vụ, làm giảm hiệu suất và trải nghiệm người dùng

- Ảnh hưởng lâu dài đến hoạt động và danh tiếng: Ngoài thiệt hại trực tiếp, một cuộc tấn công DoS có thể để lại ảnh hưởng lâu dài đến danh tiếng của công ty, khiến họ mất niềm tin từ khách hàng và đối tác. Các cuộc tấn công này có thể tái diễn, khiến doanh nghiệp phải luôn trong trạng thái sẵn sàng đối phó [2.2]

## CHƯƠNG 3 - CÁC KỸ THUẬT TẤN CÔNG DOS/DDOS PHỐ BIỀN

### 3.1 Ping of Death

#### 3.1.1 Khái niệm

Ping of Death là một loại tấn công từ chối dịch vụ, trong đó kẻ tấn công gửi một gói tin ICMP (Internet Control Message Protocol) bị dị dạng hoặc có kích thước quá khổ đến máy nạn nhân [3.1]

- Bản chất: Lợi dụng lỗ hổng trong cách hệ điều hành xử lý các gói tin IP bị phân mảnh (IP fragmentation) [3.3]

- Mục tiêu: là làm máy nạn nhân bị treo, reboot, crash dịch vụ hoặc hoạt động không ổn định khi xử lý các gói bị lỗi/oversized này [3.2]

#### 3.1.2 Cơ chế hoạt động

##### a. Giới hạn kích thước gói tin:

- Chuẩn IPv4 quy định kích thước tối đa của một gói IP là 65.535 byte (tính cả header và payload)

- Nguyên lý lỗi: Nếu tổng kích thước gói tin sau khi hệ thống nạn nhân tái lắp ghép (reassembly) vượt quá giới hạn này mà không có cơ chế kiểm tra (validation) hợp lý, sẽ sinh ra lỗi tràn bộ đệm (buffer overflow) [3.4]

##### b. Cơ chế hoạt động:

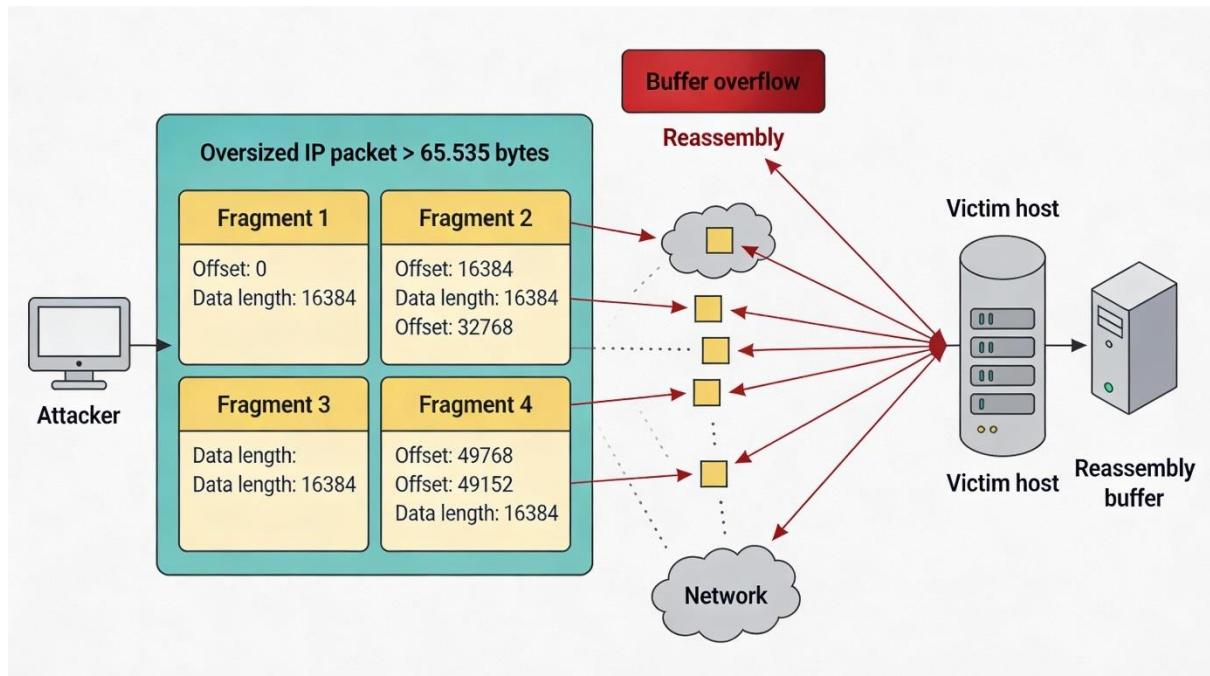
1. Tạo gói tin "ảo" quá khổ: Kẻ tấn công xây dựng một gói ICMP Echo Request (hoặc gói IP khác) sao cho khi được tái lắp ghép tại nạn nhân, kích thước logic của nó vượt 65.535 byte

2. Phân mảnh (Fragmentation): Do MTU đường truyền (ví dụ Ethernet ~1500 byte) nhỏ, gói logic này được chia thành nhiều mảnh IP (IP fragments), mỗi mảnh có kích thước hợp lệ và mang theo trường Fragment Offset

3. Gửi tới nạn nhân: Các mảnh này được gửi lần lượt tới máy nạn nhân, các thiết bị trung gian vẫn coi chúng là hợp lệ và chuyển tiếp bình thường

4. Tái lập (Reassembly) và lỗi:

- Máy nạn nhận nhận các mảnh, dựa vào Fragment Offset để ghép lại trong bộ đệm
- Do các trường offset và chiều dài được thiết kế ác ý, tổng kích thước gói sau khi ghép vượt quá 65.535 byte hoặc vượt quá vùng bộ nhớ cấp phát [3.5]
- Nếu hệ điều hành không kiểm tra điều kiện an toàn trước khi cấp phát/ghi dữ liệu, việc ghi tràn ra ngoài vùng đệm có thể làm hỏng cấu trúc dữ liệu nội bộ, dẫn đến kernel panic, treo máy hoặc khởi động lại



Hình 3.1 Mô hình tấn công Ping of Death

### 3.1.3 Tính chất

- Ở mức DoS(Tấn công đơn lẻ): chỉ cần một nguồn tấn công gửi lặp lại các gói Ping of Death cũng có thể làm các hệ điều hành hoặc thiết bị mạng cũ, chưa được vá, rơi vào trạng thái treo hoặc crash

- Ở mức DDoS(Tấn công phân tán): nhiều máy (botnet) cùng gửi các gói PoD, vừa khai thác lỗ phán mảnh, vừa tiêu tốn băng thông và tài nguyên xử lý, gây gián đoạn nghiêm trọng cho dịch vụ mạng, kể cả khi kernel không bị crash ngay

### 3.1.4 Tình trạng hiện nay và biến thể

- Tình trạng khắc phục: Hầu hết các hệ điều hành hiện đại (Windows 10/11, Linux Kernel mới, macOS) và thiết bị phần cứng sản xuất sau năm 1998 đã được vá lỗi này [3.6]. Cơ chế bảo vệ sẽ tự động loại bỏ (drop) các gói tin IP có tổng kích thước tái lập lớn hơn 65.535 bytes - Biến thể

- Ping of Death v6: Tuy tương tự tấn công dựa trên phán mảnh và xử lý ICMP không an toàn cũng từng xuất hiện ở môi trường IPv6, với một số lỗi hỏng trong việc xử lý ICMPv6 và các header mở rộng. Các hãng như Microsoft đã phải phát hành bản vá cho các lỗi

tương tự PoD trên IPv6, cho thấy kiểu lỗi này vẫn có thể tái diễn nếu việc kiểm tra kích thước và phân mảnh không được lập trình cẩn thận [3.7]

- Biến thể - Teardrop Attack: Teardrop là một dạng tấn công phân mảnh IP khác, có ý tưởng gần với Ping of Death nhưng không tập trung vào việc vượt giới hạn kích thước mà vào việc tạo các fragment có Fragment Offset chồng lấn hoặc không nhất quán. Khi hệ điều hành cố ghép các mảnh có offset "đè" lên nhau, một số implementation cũ xử lý sai, dẫn đến lỗi bộ nhớ và có thể làm hệ thống sập [3.9]

**3.1.5 Cách phòng chống:** Dù Ping of Death cổ điển hầu như không còn hiệu quả trên các hệ thống đã vá, các nguyên tắc sau vẫn rất quan trọng để phòng chống PoD và các tấn công dựa trên phân mảnh tương tự:

- Cập nhật hệ điều hành và firmware:

+ Luôn cập nhật bản vá bảo mật mới nhất cho server và thiết bị mạng [3.2]

+ Ưu tiên cập nhật các bản vá liên quan đến TCP/IP stack, ICMP/ICMPv6 và xử lý fragmentation

- Cấu hình tường lửa và thiết bị bảo mật:

+ Thiết lập tường lửa hoặc IDS/IPS để phát hiện và chặn các gói IP bị phân mảnh bất thường, ví dụ khi tổng "offset \* 8 + chiều dài dữ liệu" vượt quá giới hạn hợp lệ, hoặc có pattern fragment lạ [3.8]

+ Trong những môi trường không cần fragmentation từ Internet, có thể cân nhắc hạn chế hoặc chặn fragment ở biên, nhưng cần đánh giá kỹ để tránh ảnh hưởng các giao thức hợp lệ

- Giới hạn ICMP từ Internet:

+ Áp dụng rate limiting đối với ICMP từ bên ngoài, hoặc chỉ cho phép ICMP từ các nguồn tin cậy (ví dụ dải IP giám sát, quản trị)

+ Chỉ chặn hoàn toàn ICMP Echo Request từ Internet nếu đã cân nhắc nhu cầu giám sát và chẩn đoán; cách này giảm bớt tấn công nhưng cũng làm mất một số công cụ kiểm tra kết nối

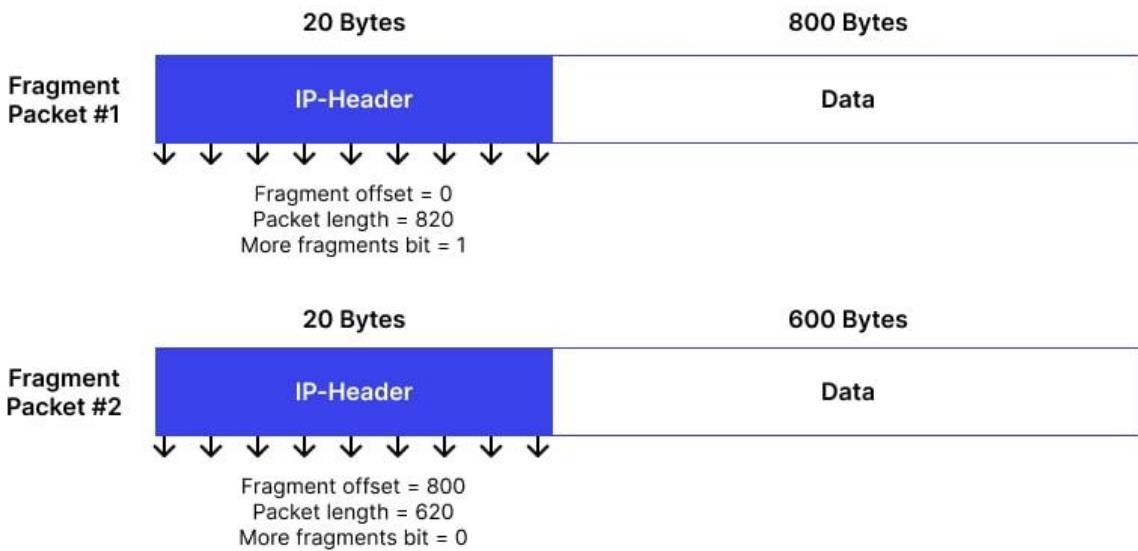
## 3.2 Teardrop Attack

### 3.2.1 Khái niệm

Teardrop Attack là một kỹ thuật tấn công từ chối dịch vụ (DoS) tinh vi, nhắm vào lỗ hổng trong cơ chế tái lắp ghép (reassembly) các gói tin IP bị phân mảnh [9] của hệ điều hành mục tiêu

- Bản chất: Khác với Ping of Death (tấn công bằng kích thước quá khổ), Teardrop tấn công bằng cách gửi các mảnh gói tin IP có thông tin định vị (offset) bị chồng chéo lên nhau (overlapping fragments)

- Mục tiêu: Làm hệ điều hành bị rối loạn, dẫn đến lỗi bộ nhớ hoặc sập hệ thống (crash/BSOD) [3.9]



Hình 3.2 Minh họa cơ chế phân mảnh IP trong tấn công Teardrop

### 3.2.2 Cơ chế hoạt động:

- Quy trình phân mảnh IP thông thường: Khi một gói tin lớn được chia nhỏ, mỗi mảnh sẽ có một trường Fragment Offset để báo cho máy nhận biết mảnh đó nằm ở vị trí nào trong gói tin gốc

- Giới hạn và đặc điểm phân mảnh IP:

- + Mỗi mảnh IP có trường Fragment Offset (13 bit, đơn vị 8 byte) và cờ More Fragments (MF) để chỉ vị trí và thứ tự ghép
- + Hệ điều hành phải xử lý đúng các trường hợp offset hợp lệ, không overlap; nếu không, quá trình reassembly có thể ghi đè bộ nhớ hoặc gây exception

- Cơ chế hoạt động:

1. Tạo các mảnh overlap: Kẻ tấn công xây dựng nhiều mảnh IP từ một gói gốc "áo", nhưng cố tình đặt Fragment Offset chồng lấn (ví dụ Fragment 1: offset 0–1000, Fragment 2: offset 500–1500, Fragment 3: offset 1000–2000)
2. Đánh dấu MF sai lệch: Các mảnh có cờ MF và offset không nhất quán, khiến tổng kích thước hoặc thứ tự ghép bị mơ hồ
3. Gửi tới nạn nhân: Các mảnh riêng lẻ vẫn hợp lệ về kích thước (vừa MTU), nên qua được router trung gian và tới host đích
4. Tái lắp ghép và phát sinh lỗi [3.10]:
  - Máy nạn nhân nhận mảnh, cố ghép dựa trên offset nhưng gặp xung đột (overlap hoặc hole)

- Các stack IP cũ (như Windows NT 4.0, Linux kernel <2.0.39) không xử lý overlap đúng cách, dẫn đến ghi đè bộ nhớ, infinite loop hoặc kernel panic
- Kết quả: hệ thống treo, reboot hoặc crash dịch vụ

### 3.2.3 Tính chất

- Tấn công dựa trên giao thức (Protocol-based): Teardrop khai thác lỗi logic trong việc triển khai ngăn xếp giao thức TCP/IP, không phụ thuộc vào việc làm nghẽn băng thông
- Hiệu quả cao trên hệ thống cũ: Đối với các hệ thống dính lỗ hỏng này, chỉ cần một vài gói tin Teardrop được chế tạo kỹ lưỡng là đủ để gây sập hệ thống (kiểu tấn công "bắn tỉa")

### 3.2.4 Tình trạng hiện nay và biến thể

- Hầu hết hệ điều hành hiện đại (Windows hiện hành, Linux kernel mới, BSD) đã vá bằng cách kiểm tra và loại bỏ fragment overlap hoặc malformed trong reassembly, tấn công chỉ hiệu quả với hệ cũ hoặc thiết bị nhúng chưa cập nhật
- Teardrop thường được coi là "anh em" của Ping of Death vì cùng khai thác fragmentation, nhưng tập trung vào overlap thay vì oversized. Các biến thể như New Tear hoặc Land Attack kết hợp overlap với spoof IP để tăng độ khó phát hiện và khai thác thêm lỗ hỏng khác

**3.2.5 Cách phòng chống:** Dù Teardrop có diễn ít gặp trên hệ thống mới, các biện pháp sau vẫn cần thiết cho fragmentation attack nói chung

- Cập nhật bản vá hệ thống: Đây là biện pháp quan trọng nhất. Đảm bảo tất cả các máy chủ và thiết bị mạng đang chạy các phiên bản hệ điều hành ổn định, đã được vá các lỗ liên quan đến xử lý phân mảnh IP

- Cấu hình Firewall và IPS (Hệ thống ngăn chặn xâm nhập) [3.10, 3.11]:

+ Stateful Inspection: Sử dụng tường lửa có khả năng kiểm tra trạng thái (Stateful Firewall). Tường lửa sẽ thực hiện tái lắp ghép ảo các mảnh gói tin trước khi cho phép chúng đi vào mạng nội bộ. Nếu phát hiện các offset chồng chéo, tường lửa sẽ loại bỏ toàn bộ gói tin

+ Chặn gói tin phân mảnh dị dạng: Cấu hình các thiết bị bảo mật biên để tự động loại bỏ (drop) bất kỳ gói tin IP nào có cờ báo phân mảnh nhưng các thông số offset không hợp lệ

+ Áp dụng Deep Packet Inspection (DPI) để tái lắp ghép và kiểm tra fragment trước khi forward tới host nội bộ

- Giám sát và giới hạn fragmentation:

+ Giám sát lưu lượng IP fragment bắt thường qua công cụ như Wireshark, Snort hoặc SIEM; cảnh báo khi tỷ lệ fragment tăng đột biến

+ Giới hạn hoặc drop fragment từ Internet nếu không cần thiết, nhưng test kỹ để tránh ảnh hưởng VPN, tunnel hoặc giao thức lớn gói (Path MTU Discovery)

- Tắt chức năng tái lắp ghép trên Router biên: Trên các Router ở lớp biên mạng, nếu không cần thiết, có thể cấu hình để không thực hiện việc tái lắp ghép các gói tin phân mảnh, giảm thiểu rủi ro cho chính thiết bị đó

### 3.3 TCP SYN Flood

#### 3.3.1 Khái niệm:

TCP SYN Flood (hay còn gọi là SYN Flooding) là một hình thức tấn công từ chối dịch vụ (DoS) phổ biến, nhắm vào quy trình bắt tay 3 bước (3-way handshake) của giao thức TCP [3.12, 3.13]

- Bản chất: Kẻ tấn công gửi liên tiếp các yêu cầu kết nối (SYN) giả mạo đến máy chủ mục tiêu nhưng không bao giờ hoàn tất quá trình kết nối

- Mục tiêu: Làm cạn kiệt tài nguyên của máy chủ (cụ thể là bảng trạng thái kết nối - connection table), khiến máy chủ không thể tiếp nhận thêm bất kỳ kết nối hợp lệ nào từ người dùng thực

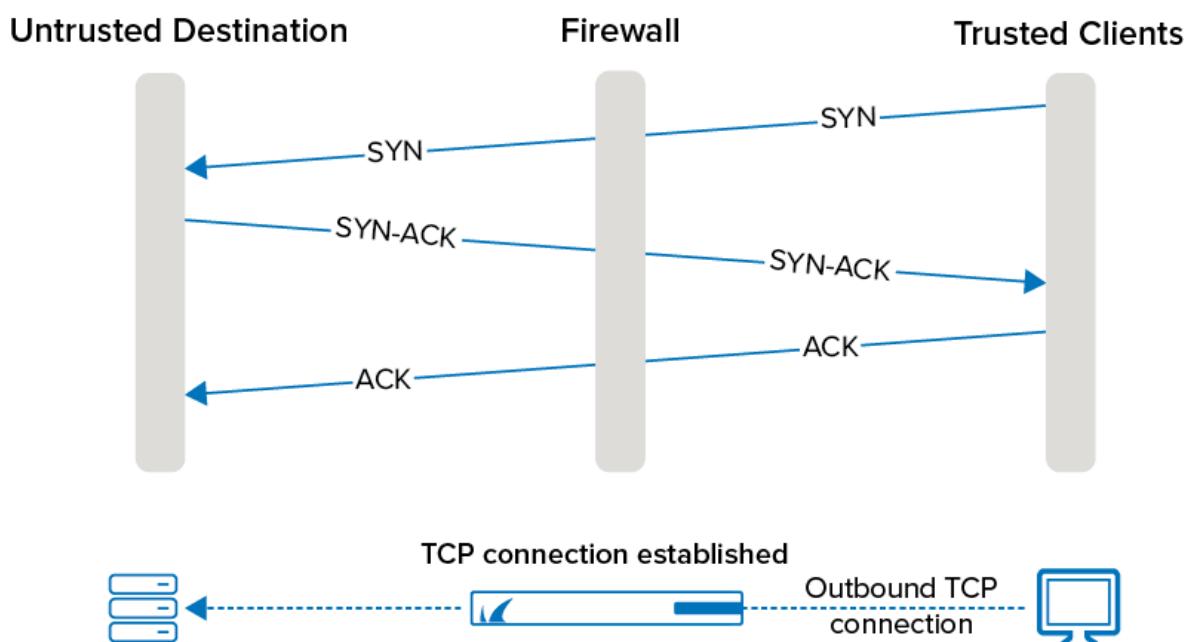
#### 3.3.2 Cơ chế hoạt động

##### a. Quy trình bắt tay 3 bước chuẩn:

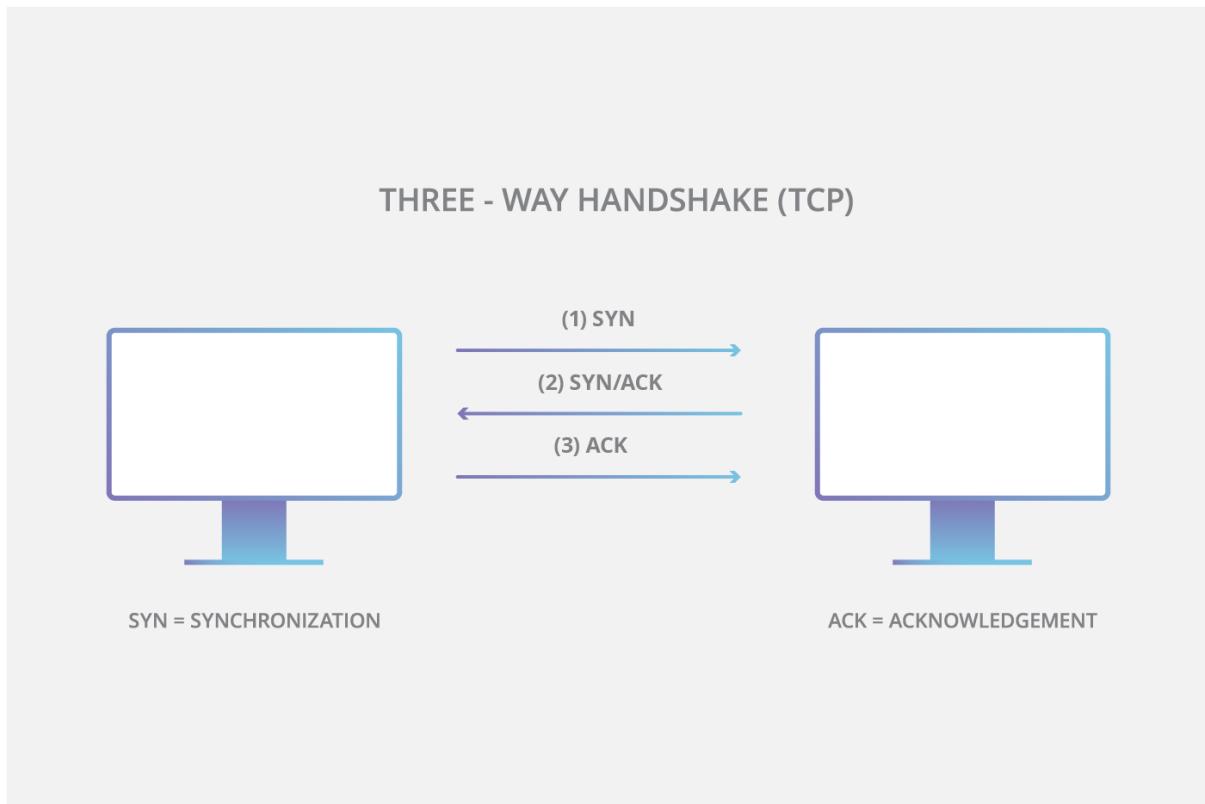
- Bước 1 (SYN): Client gửi gói tin SYN (Synchronize) để yêu cầu kết nối

- Bước 2 (SYN-ACK): Server nhận được, phản hồi bằng gói SYN-ACK và dành một phần bộ nhớ (trong hàng đợi Backlog Queue) để lưu trạng thái chờ

- Bước 3 (ACK): Client gửi lại gói ACK để xác nhận. Kết nối được thiết lập xong



Hình 3.3 Quy trình bắt tay 3 bước (3-way handshake) của giao thức TCP



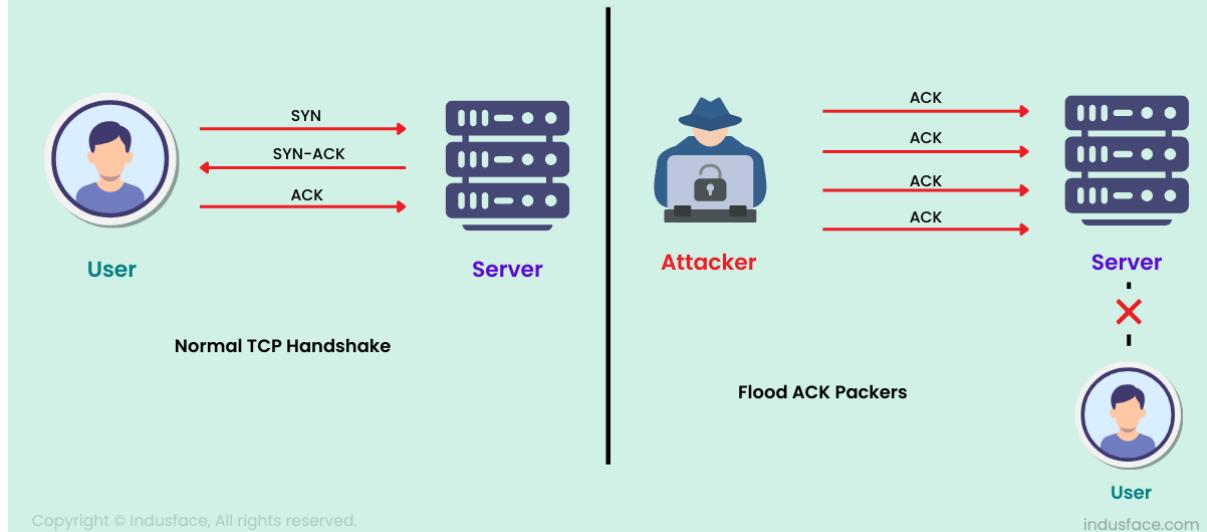
*Hình 3.4 Các bước trong quy trình bắt tay 3 bước (3-Way Handshake)*

### b. Quy trình tấn công SYN Flood:

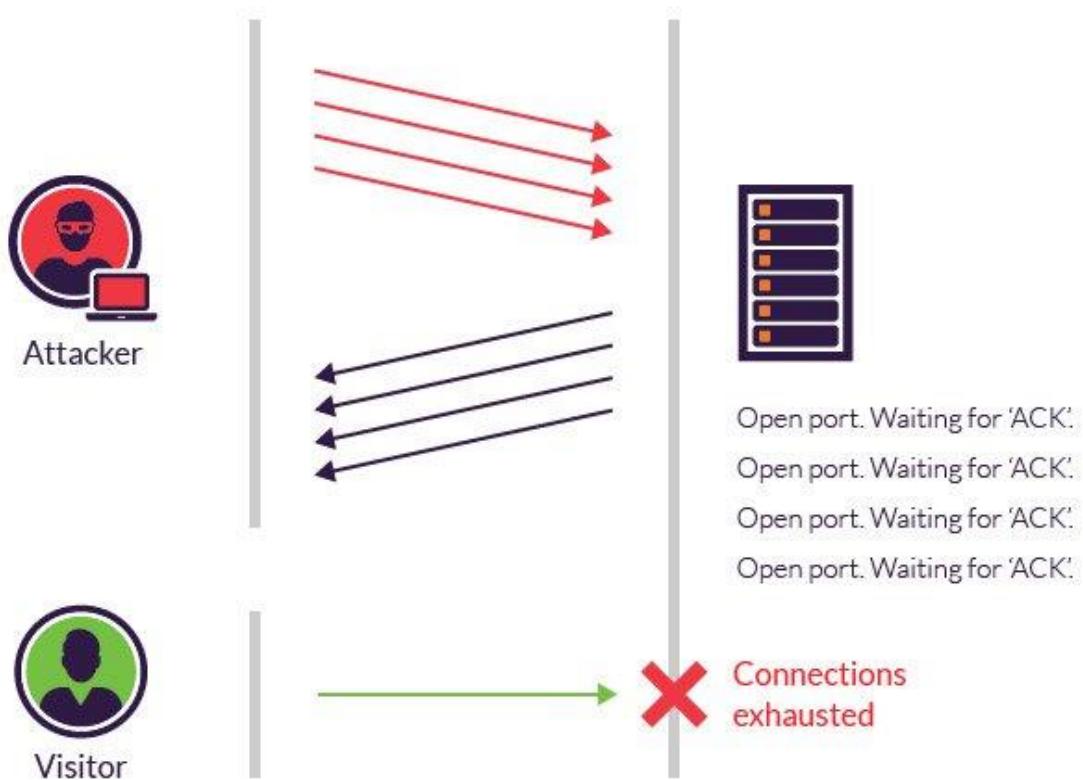
- Gửi ồ ạt: Kẻ tấn công gửi hàng loạt gói tin SYN đến Server. Thường các gói này sử dụng địa chỉ IP nguồn giả mạo (IP Spoofing) [3.14]
- Server mắc bẫy: Với mỗi gói SYN nhận được, Server tưởng là yêu cầu thật, nó phản hồi SYN-ACK và cấp phát bộ nhớ để lưu trạng thái "Mở một nửa" (Half-open connection)
- Bỏ rơi:
  - + Vì địa chỉ IP nguồn là giả, gói SYN-ACK của Server gửi đi sẽ không đến được đích hoặc bị máy đích (nếu có) hủy bỏ
  - + Do đó, Server không bao giờ nhận được gói ACK phản hồi
- Cạn kiệt tài nguyên: Server vẫn phải duy trì các kết nối "mở một nửa" này trong bộ nhớ cho đến khi hết thời gian chờ (Timeout). Khi hàng ngàn yêu cầu giả mạo lấp đầy hàng đợi (Backlog Queue), Server không còn chỗ trống để tiếp nhận các gói SYN từ người dùng hợp lệ → Từ chối dịch vụ [3.15]

### TCP ACK Flood Attack

INDUSFACE™



Hình 3.5 Mô hình tấn công TCP ACK Flood



Hình 3.6 Cơ chế tấn công SYN Flood và trạng thái kết nối mở một nửa (Half-open)

#### 3.3.3 Tính chất

- Là tấn công ở tầng giao vận (Transport layer), nhắm trực tiếp vào việc quản lý phiên TCP của server

- Dễ mở rộng thành DDoS khi nhiều máy trong botnet cùng gửi SYN đến một hoặc nhiều cổng dịch vụ (thường là 80, 443, 22,...)

- Tấn công bất đối xứng (Asymmetric): Kẻ tấn công tốn rất ít tài nguyên (chỉ cần gửi gói tin nhỏ), trong khi nạn nhân tốn nhiều tài nguyên (phải cấp phát bộ nhớ và duy trì trạng thái chờ)

- IP Spoofing (Giả mạo IP): Hầu hết các cuộc tấn công SYN Flood đều dùng IP giả. Điều này giúp kẻ tấn công ẩn danh và quan trọng hơn là tránh việc máy của chính kẻ tấn công gửi gói RST (Reset) để đóng kết nối, làm hỏng cuộc tấn công

### 3.3.4 Tình trạng hiện nay và biến thể

- Mức độ phổ biến: Đây vẫn là một trong những kỹ thuật DDoS phổ biến nhất thế giới vì

+ Tương đối dễ sinh lưu lượng (dùng spoof IP, không cần đáp lại)

+ Nhiều dịch vụ Internet vẫn dựa trên TCP (web, mail, SSH, API...)

- Biến thể - SYN-ACK Flood: Thay vì gửi SYN, kẻ tấn công gửi gói SYN-ACK (giả vờ như Server đang phản hồi Client) để làm rối loạn bảng trạng thái hoặc làm nghẽn tường lửa

- Tấn công kết hợp với các lỗi cấu hình TCP (backlog nhỏ, timeout dài)

- Tình trạng khắc phục: Các thiết bị tường lửa hiện đại và hệ điều hành ngày nay đã có khả năng chống chịu tốt hơn nhờ các thuật toán thông minh, nhưng vẫn có thể bị đánh sập nếu lưu lượng tấn công (pps - packets per second) quá lớn vượt quá khả năng xử lý của phần cứng

### 3.3.5 Cách phòng chống:

Để giảm thiểu tác động của SYN Flood, cần áp dụng mô hình bảo vệ đa lớp từ cấu hình máy chủ đến hạ tầng mạng:

- Cấu hình hệ điều hành (TCP stack) Can thiệp vào các thông số Kernel để tối ưu hóa khả năng xử lý kết nối:

+ Giảm thời gian chờ (SYN-RECEIVED timeout): Rút ngắn thời gian Server chờ gói ACK phản hồi. Nếu quá thời gian này mà không nhận được ACK, Server sẽ hủy kết nối ngay lập tức để giải phóng bộ nhớ + Tăng kích thước hàng đợi SYN backlog nếu tài nguyên cho phép [16]

+ Tăng kích thước hàng đợi (SYN Backlog): Mở rộng giới hạn hàng đợi chứa các kết nối đang chờ (pending connections) nếu tài nguyên RAM cho phép, giúp Server chịu được lượng request lớn hơn

+ Bật các cơ chế bảo vệ đặc thù: Kích hoạt tính năng SYN Cookies, Syncache, hoặc Synproxy (tùy thuộc vào Hệ điều hành). Các cơ chế này cho phép xác thực kết nối trước khi cấp phát bộ nhớ thực sự [3.17, 3.18]

- Tường lửa và thiết bị bảo vệ sử dụng Firewall thế hệ mới (NGFW), IPS hoặc Load Balancer làm lớp chắn trước Server:

+ Giới hạn tốc độ (Rate Limiting): Cấu hình giới hạn số lượng gói tin SYN được phép gửi đến từ một nguồn hoặc tới một đích trong một khoảng thời gian nhất định

+ Lọc SYN bất thường: Thiết lập quy tắc chặn các gói SYN đến từ các IP nằm trong danh sách đen, các dải IP lạ (bogus IP), hoặc có các mẫu (pattern) header đáng ngờ

+ Chế độ "SYN Proxy": Cấu hình thiết bị bảo mật hoạt động như một trung gian. Thiết bị này sẽ trả lời SYN-ACK thay cho Server. Chỉ khi nào quá trình bắt tay hoàn tất (nhận được ACK hợp lệ), thiết bị mới chuyển kết nối đó tới Server thật [3.19, 3.20]

- Kiến trúc và hạ tầng:

+ Sử dụng Proxy/CDN: Đặt dịch vụ web sau các hệ thống Reverse Proxy (như Nginx, HAProxy) hoặc mạng phân phối nội dung (CDN). Các hệ thống này có hạ tầng mạnh mẽ để hấp thụ và phân tán lưu lượng tấn công thay cho Server gốc

+ Dịch vụ chống DDoS chuyên dụng: Đối với các hệ thống quan trọng đối mặt nguy cơ tấn công lớn, cần sử dụng các dịch vụ Scrubbing Center (Trung tâm làm sạch lưu lượng) từ các nhà cung cấp ISP hoặc Cloud để lọc bỏ lưu lượng độc hại trước khi nó đến được mạng của tổ chức

### 3.4 DNS Amplification Attack

#### 3.4.1 Khái niệm

DNS Amplification Attack (Tấn công khuếch đại DNS) là một dạng tấn công từ chối dịch vụ phân tán (DDoS) dựa trên cơ chế phản xạ (Reflection-based) [3.21]

- Bản chất: Kẻ tấn công lợi dụng các máy chủ DNS mở (Open DNS Resolvers) công khai trên Internet để làm ngập lụt băng thông của nạn nhân

- Mục tiêu: Làm nghẽn đường truyền mạng của nạn nhân bằng lượng dữ liệu phản hồi khổng lồ, khiến người dùng hợp lệ không thể truy cập dịch vụ

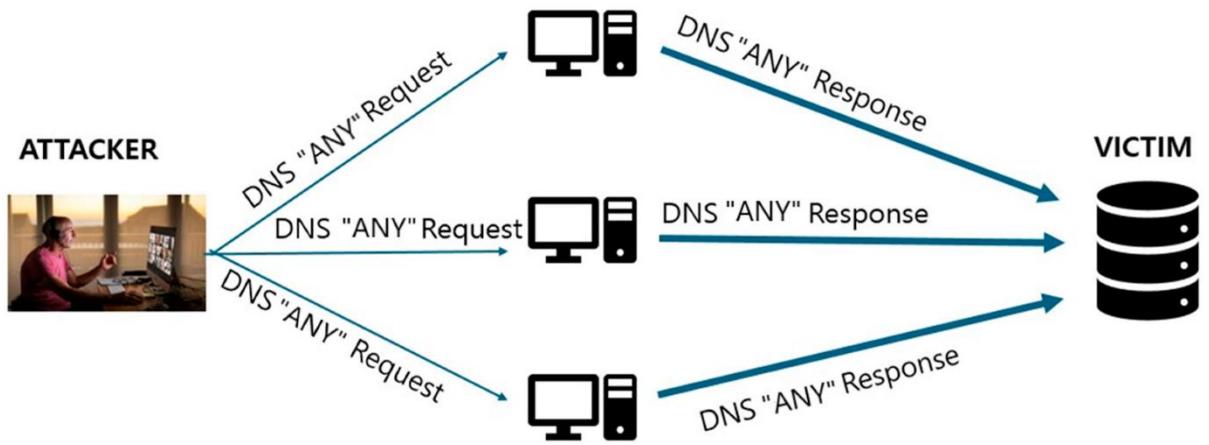
- Nguyên lý lõi:

+ Giả mạo (Spoofing): Giả danh nạn nhân.

+ Khuếch đại (Amplification): Gửi yêu cầu nhỏ nhưng nhận về phản hồi cực lớn

#### 3.4.2 Cơ chế hoạt động

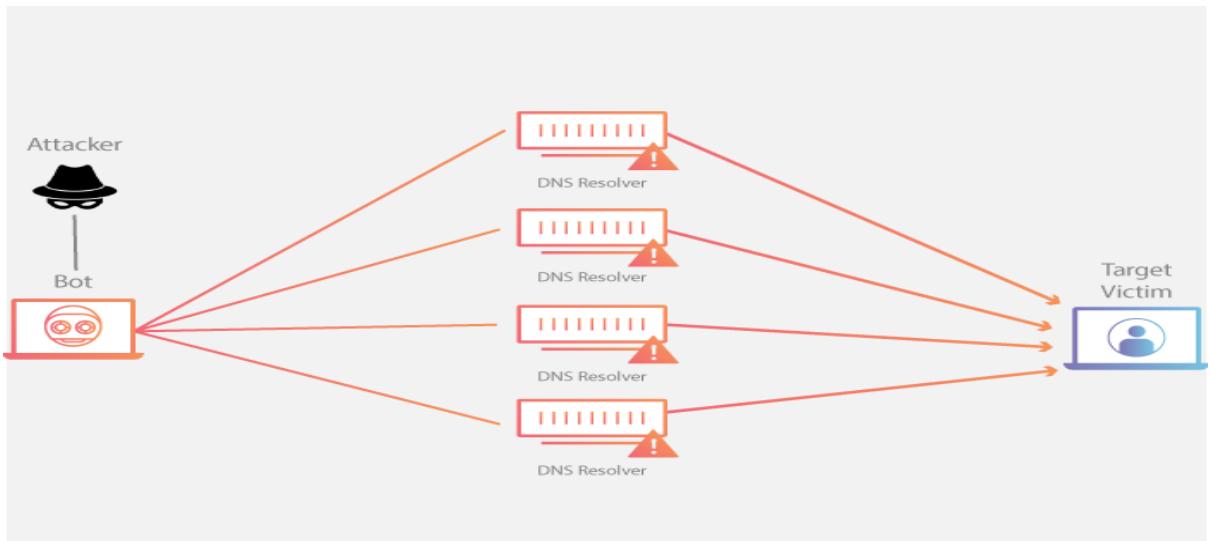
- Kỹ thuật này khai thác giao thức UDP (User Datagram Protocol) - một giao thức không yêu cầu thiết lập kết nối (connectionless), cho phép kẻ tấn công dễ dàng giả mạo địa chỉ IP nguồn



Hình 3.7 Mô hình tấn công khuéch đại DNS (DNS Amplification Attack)

- Quy trình tấn công:

1. Chọn máy chủ khuéch đại (Reconnaissance): Attacker dò tìm hoặc sử dụng danh sách có sẵn các Open DNS Resolvers (các máy chủ DNS được cấu hình sai, chấp nhận truy vấn để quy từ bất kỳ địa chỉ IP nào trên Internet)
2. Giả mạo địa chỉ nguồn (IP Spoofing): Attacker gửi hàng loạt gói tin DNS Query đến các Open Resolvers này. Tuy nhiên, địa chỉ IP nguồn trong gói tin không phải là IP của Attacker, mà đã bị giả mạo (spoofed) thành địa chỉ IP của nạn nhân
3. Khuéch đại (Amplification):
  - Các truy vấn được thiết kế đặc biệt để yêu cầu phản hồi có kích thước lớn nhất có thể (ví dụ: dùng truy vấn type ANY để lấy toàn bộ bản ghi, hoặc truy vấn các vùng có sử dụng DNSSEC) [3.22]
  - Mỗi máy chủ DNS lúc này trở thành một "máy khuéch đại". Chúng nhận truy vấn nhỏ nhưng trả lời bằng gói tin kích thước rất lớn về phía địa chỉ IP của nạn nhân
4. Nạn nhân bị ngập lụt (Flooding): Nạn nhân, dù không gửi yêu cầu nào, bỗng nhiên phải hứng chịu vô số phản hồi DNS không mong muốn từ khắp nơi đổ về. Điều này dẫn đến tắc nghẽn đường truyền Internet hoặc quá tải CPU/Stack mạng của các thiết bị tường lửa/máy chủ



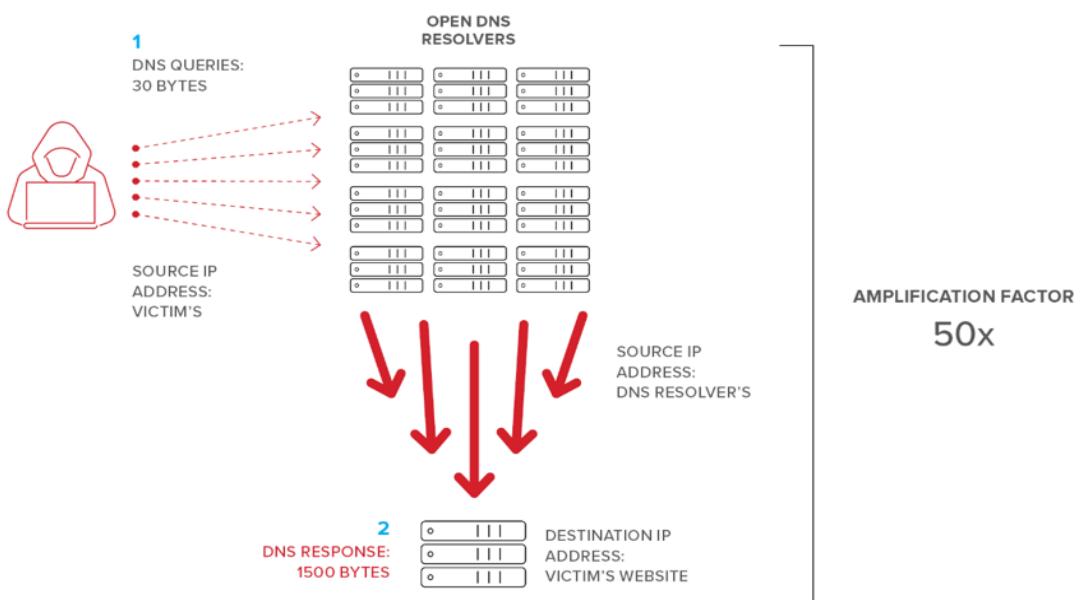
Hình 3.8 Sơ đồ luồng tấn công DNS Amplification qua các Open Resolvers

### 3.4.3 Tính chất

Mặc dù sử dụng giao thức ở tầng ứng dụng (Application Layer - DNS), nhưng tác động của kỹ thuật này lại mang tính chất của tấn công hạ tầng (Infrastructure Attack)

- Reflection (Tính phản xạ): Các máy chủ DNS đóng vai trò trung gian. Kẻ tấn công không trực tiếp liên lạc với nạn nhân, giúp chúng ẩn danh tốt hơn và khó bị truy vết nguồn gốc thực sự

- Amplification (Tính khuếch đại): Đây là đặc điểm nguy hiểm nhất. Kẻ tấn công biến một lượng băng thông nhỏ (gói tin truy vấn ~60 bytes) thành một lượng băng thông khổng lồ tấn công nạn nhân (gói tin phản hồi > 3000 bytes). Hệ số khuếch đại có thể lên tới 50 lần hoặc 70 lần [3.22]



### *Hình 3.9 Minh họa hệ số khuếch đại (Amplification Factor) trong tấn công DNS*

#### **3.4.4 Tình trạng hiện nay và biến thể**

- Tình trạng: DNS Amplification vẫn là một trong những kỹ thuật DDoS phổ biến nhất hiện nay do DNS là dịch vụ nền tảng của Internet và số lượng Open Resolvers không được bảo mật vẫn còn rất lớn

- Biến thể và chiêu trò:

+ Kết hợp với Botnet: Kẻ tấn công sử dụng mạng máy tính ma (Botnet) để gửi truy vấn, tạo ra cuộc tấn công từ nhiều hướng, khiến việc lọc IP trở nên vô cùng khó khăn

+ Tấn công đa vector (Multi-vector DDoS): Kẻ tấn công kết hợp DNS Amplification với các giao thức khuếch đại khác (như NTP, SSDP, CLDAP...) để tấn công đồng thời, làm tê liệt các hệ thống phòng thủ đơn lẻ

#### **3.4.5 Cách phòng chống**

- Ở phía máy chủ DNS (Ngăn chặn bị lợi dụng):

+ Không vận hành Open Resolver: Cấu hình máy chủ DNS chỉ trả lời truy vấn từ mạng nội bộ hoặc các dải IP khách hàng được ủy quyền

+ Rate Limiting (Giới hạn tốc độ): Cấu hình giới hạn số lượng phản hồi DNS cho một địa chỉ IP cụ thể trong một đơn vị thời gian (Response Rate Limiting - RRL)

+ Hạn chế truy vấn lớn: Tắt hoặc hạn chế các truy vấn loại ANY nếu không cần thiết để giảm thiểu kích thước gói tin phản hồi

- Ở phía mạng trung gian / ISP (Ngăn chặn nguồn gốc):

+ Lọc địa chỉ nguồn (Ingress Filtering - BCP38): Các nhà cung cấp dịch vụ (ISP) áp dụng chuẩn BCP38 để chặn các gói tin giả mạo IP (spoofed IP) ngay từ biên mạng của khách hàng. Nếu không thể giả IP, tấn công phản xạ sẽ vô hiệu [3.22]

+ Hệ thống lọc tập trung: Triển khai các hệ thống phát hiện và lọc DDoS tại core network để chặn các mẫu lưu lượng DNS bất thường trước khi chúng đến đích

- Ở phía nạn nhân / Hệ thống đích (Giảm thiểu thiệt hại):

+ Sử dụng hạ tầng chống DDoS chuyên dụng: Thuê dịch vụ DNS và bảo mật từ các nhà cung cấp có hạ tầng Anycast và các trung tâm làm sạch lưu lượng (Scrubbing Centers) lớn [3.21]

+ Kiến trúc phòng thủ: Đặt các dịch vụ quan trọng sau các lớp bảo vệ như CDN (Content Delivery Network), Reverse Proxy hoặc thiết bị cân bằng tải có chức năng chống DDoS để hấp thụ và phân tán lưu lượng tấn công

Bảng so sánh các đặc điểm kỹ thuật, cơ chế và mức độ ảnh hưởng của các phương thức tấn công đã trình bày

Tiêu chí	Ping of Death (PoD)	Teardrop Attack	TCP SYN Flood	DNS Amplification
<b>Phân loại</b>	Tấn công phân mảnh (Fragmentation).	Tấn công phân mảnh (Fragmentation)	Tấn công giao thức (Protocol Attack)	Tấn công thể tích (Volumetric / Reflection)
<b>Tầng OSI</b>	Layer 3 (Network)	Layer 3 (Network)	Layer 4 (Transport)	Layer 7 (App) → ảnh hưởng Layer 3/4
<b>Cơ chế chính</b>	Gửi gói tin có kích thước quá khổ ( $>65.535$ bytes) sau khi tái lập	Gửi các mảnh gói tin có Offset chồng chéo lên nhau (Overlapping)	Gửi hàng loạt SYN giả mạo, bỏ dở quá trình bắt tay 3 bước	Giả mạo IP nạn nhân gửi query đến DNS Server để nhận phản hồi cực lớn
<b>Điểm yếu khai thác</b>	Lỗi xử lý bộ nhớ đệm (Buffer Overflow) của hệ điều hành	Lỗi logic trong thuật toán lắp ghép gói tin của hệ điều hành	Giới hạn bộ nhớ lưu trữ trạng thái kết nối (Backlog Queue) của Server	Khả năng khuếch đại của giao thức UDP/DNS và các Open Resolvers
<b>Mục tiêu tác động</b>	Gây sập hệ thống (Crash/BSOD), khởi động lại	Gây sập hệ thống (Crash), treo máy	Làm cạn kiệt tài nguyên (RAM/CPU), không thể kết nối mới	Làm nghẽn đường truyền (Bão hòa băng thông)
<b>Dấu hiệu nhận biết</b>	Gói tin có IP Total Length tái lập $> 65.535$ bytes	Các gói tin có Fragment Offset + Length bị trùng lắp	Tỉ lệ gói SYN cao bất thường, không có ACK phản hồi	Lưu lượng UDP port 53 tăng đột biến, gói tin phản hồi lớn
<b>Cách phòng chống</b>	Cập nhật bản vá (Patch) hệ điều hành	Cấu hình Firewall chặn gói tin phân mảnh lỗi	Bật SYN Cookies, giảm Timeout,	Chặn IP giả mạo (BCP38), Rate Limit,

			dùng Firewall/IPS	dùng dịch vụ Anti-DDoS
--	--	--	-------------------	------------------------

### Nhận xét:

#### 1. Về mục đích:

- PoD và Teardrop thiên về việc "giết chết" (Kill) máy chủ ngay lập tức bằng cách khai thác lỗi phần mềm (Bug)
- SYN Flood và DNS Amplification thiên về việc "làm tê liệt" (Exhaustion) khả năng phục vụ bằng cách làm quá tải tài nguyên hoặc đường truyền

#### 2. Về khả năng phòng thủ:

- Các lỗi như PoD và Teardrop dễ phòng chống triệt để bằng cách cập nhật phần mềm (Patching)
- Các tấn công như SYN Flood và DNS Amplification khó phòng chống hơn vì chúng khai thác kiến trúc giao thức chuẩn, đòi hỏi phải có thiết bị chuyên dụng và hạ tầng mạng đủ mạnh để lọc và chịu tải

### 3.5 Đặc điểm và xu hướng tấn công DoS/DDoS hiện nay

#### 3.5.1 Đặc điểm chung của các cuộc tấn công hiện đại

- Lưu lượng phân tán, đa nguồn (Distributed & Multi-source): Thay vì một vài máy tấn công, lưu lượng độc hại xuất phát từ các mạng botnet khổng lồ (bao gồm PC, máy chủ, và đặc biệt là hàng triệu thiết bị IoT/camera bị nhiễm mã độc). Điều này khiến việc chặn tấn công dựa trên IP đơn lẻ (IP blocking) trở nên vô hiệu

- Tấn công đa lớp, đa vectơ (Multi-vector): Tin tặc hiếm khi sử dụng một kỹ thuật duy nhất. Chúng kết hợp đồng thời:

+ Tấn công băng thông (Volumetric) như *DNS Amplification* để làm nghẽn đường truyền

+ Tấn công giao thức (Protocol) như *SYN Flood* để bao mòn tài nguyên TCP stack

+ Tấn công ứng dụng (Application) như *HTTP Flood* để đánh sập logic xử lý web [3.27]

- Tập trung vào tính sẵn sàng: Mục tiêu tối thượng là khiến dịch vụ bị gián đoạn (chậm, timeout) hoặc mất ổn định hoàn toàn, gây thiệt hại trực tiếp về kinh tế và uy tín thương hiệu

#### 3.5.2. Xu hướng chuyển dịch kỹ thuật

Tin tặc đang thay đổi chiến thuật để vượt qua các tường lửa truyền thống:

- Dịch chuyển lên Tầng ứng dụng (Layer 7):

+ Tỉ lệ tấn công vào HTTP/HTTPS, WebSocket và gRPC tăng mạnh. Các báo cáo gần đây ghi nhận tấn công Layer 7 tăng khoảng 30–40%, trong đó HTTP-based DDoS chiếm phần lớn

+ Lý do: Tấn công tầng ứng dụng mô phỏng hành vi người dùng thật, khó bị phát hiện bởi các bộ lọc mạng (Packet Filter) thông thường

- Bùng nổ tấn công vào API (Application Programming Interface):

+ Với sự phổ biến của kiến trúc Microservices, các API trở thành mục tiêu hàng đầu

+ Thống kê năm 2025 cho thấy tấn công nhắm vào API tăng khoảng 70%. Kẻ tấn công lợi dụng các truy vấn API nặng (expensive queries) để làm quá tải hệ thống backend mà không cần băng thông lớn [3.25, 3.26]

- Chiến thuật "Đánh chính xác" (Precision Attacks):

+ Thay vì "xả lũ" (flood) bừa bãi, tin tặc chuyển sang dạng tấn công "Pulse Wave" (Sóng xung kích): Các đợt tấn công ngắn, dồn dập, ngắt quãng để đánh lừa cơ chế lấy mẫu (sampling) và ngửng phát hiện của thiết bị bảo mật

### 3.5.3. Quy mô và tần suất kỷ lục

- Số lượng khổng lồ: Thế giới ghi nhận hàng triệu cuộc tấn công mỗi năm. Riêng nửa đầu năm 2025, số lượng sự kiện DDoS toàn cầu đã vượt mốc 8 triệu [24]

- Đỉnh băng thông (Peak Bandwidth): Khả năng huy động botnet và các máy chủ khuếch đại (Amplifiers) ngày càng mạnh. Các đợt tấn công đạt ngưỡng 2–3 Tbps trở nên phổ biến, cá biệt có trường hợp ghi nhận lên tới 5–7 Tbps [3.23]

### 3.5.4 Hình thức tổ chức và động cơ

- DDoS-as-a-Service (DDoS như một dịch vụ): Sự xuất hiện của các dịch vụ "Booter" hay "Stresser" cho phép bất kỳ ai (kể cả người không có kỹ năng) cũng có thể thuê hệ thống botnet để tấn công đối thủ với chi phí rẻ, tính theo giờ hoặc ngày

- Động cơ địa chính trị: Tấn công DDoS ngày càng gắn liền với các xung đột chính trị hoặc quân sự (Cyberwarfare), nhắm vào hạ tầng trọng yếu của quốc gia như chính phủ, tài chính, viễn thông và năng lượng

### 3.5.5 Xu hướng các Vector tấn công cụ thể

- Amplification/Reflection vẫn chiếm ưu thế: Các giao thức UDP không kết nối tiếp tục bị lạm dụng. Trong đó, DNS Amplification (như đã trình bày ở mục 3.4) vẫn chiếm hơn 50% lưu lượng tấn công khuếch đại. Ngoài ra còn có NTP, CLDAP, SSDP

- Xuất hiện các Vector mới:

+ Lợi dụng giao thức mới: QUIC, HTTP/2 Rapid Reset

+ Khai thác lỗ hổng cấu hình: IPv6 reflection, WebRTC/TURN

+ Memcached amplification: Một biến thể nguy hiểm với hệ số khuếch đại lên tới hàng chục nghìn lần (51.000x)

### 3.5.6 Tổng kết: So sánh kỹ thuật cổ điển và hiện đại

<b>Tiêu chí</b>	<b>Tấn công cổ điển</b> (VD: PoD, Teardrop)	<b>Tấn công hiện đại</b> (VD: HTTP Flood, API Attack, DNS Amp)
<b>Bản chất</b>	Khai thác lỗ lây trình/xử lý của Hệ điều hành (Bugs)	Khai thác giới hạn tài nguyên hoặc logic ứng dụng
<b>Quy mô</b>	Rất nhỏ (vài gói tin dạng)	Rất lớn (Volumetric - Tbps) hoặc Rất tinh vi (L7)
<b>Mục tiêu</b>	Làm sập hệ thống (Crash/BSOD)	Làm nghẽn đường truyền hoặc cạn kiệt tài nguyên xử lý
<b>Cách phòng chống</b>	Cập nhật bản vá (Patching) là xong	Cần hệ thống phòng thủ đa lớp (Firewall, IPS, CDN, WAF, Scrubbing Center)
<b>Độ phức tạp</b>	Đơn vector	Đa vector, kết hợp Botnet toàn cầu

## CHƯƠNG 4 - CÁCH THỨC TẤN CÔNG DOS/DDOS VÀ MÔ HÌNH BONET

Chương này tập trung mô tả cách thức triển khai các cuộc tấn công DoS/DDoS trong thực tế, đặc biệt nhấn mạnh mô hình botnet và quy trình điều phối tấn công. Các phân tích chi tiết về cơ chế kỹ thuật và điểm yếu giao thức của từng phương thức tấn công đã được trình bày ở Chương 3, nội dung trong chương này nhằm minh họa luồng tấn công, các thành phần tham gia và tác động thực tế đối với hệ thống mục tiêu

### 4.1 Cách thức tấn công DoS

DoS (Denial of Service) là hình thức tấn công từ một nguồn duy nhất nhằm làm gián đoạn hoạt động của hệ thống mục tiêu bằng cách làm cạn kiệt tài nguyên

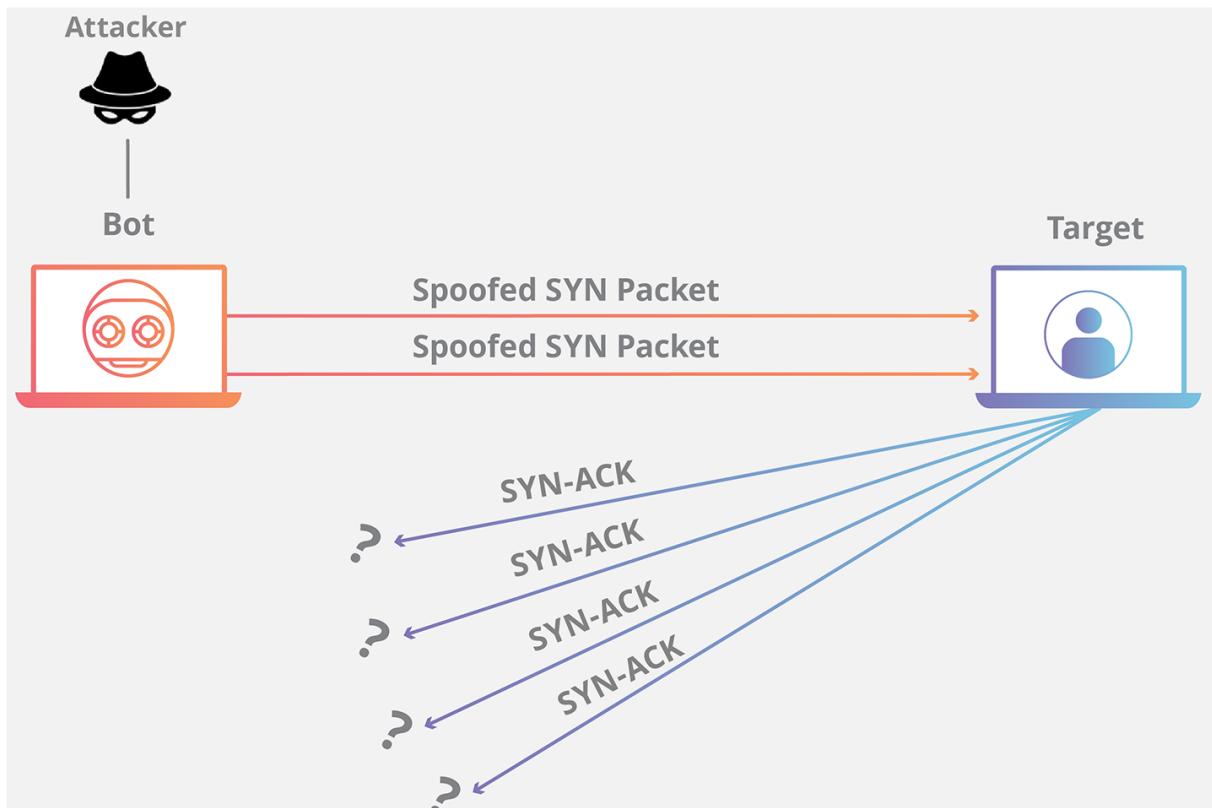
#### 4.1.1 Tấn công SYN Flood

Trong kịch bản SYN Flood, kẻ tấn công gửi liên tục các yêu cầu thiết lập kết nối TCP đến máy chủ nhưng không hoàn tất quá trình kết nối, khiến tài nguyên xử lý kết nối của máy chủ bị chiếm dụng

Ví dụ: Một máy tính tấn công gửi hàng nghìn gói SYN mỗi giây đến cổng dịch vụ web của máy chủ

Kết quả:

- Băng kết nối TCP bị đầy
- Máy chủ không tiếp nhận được kết nối hợp lệ
- Dịch vụ web bị treo hoặc phản hồi chậm



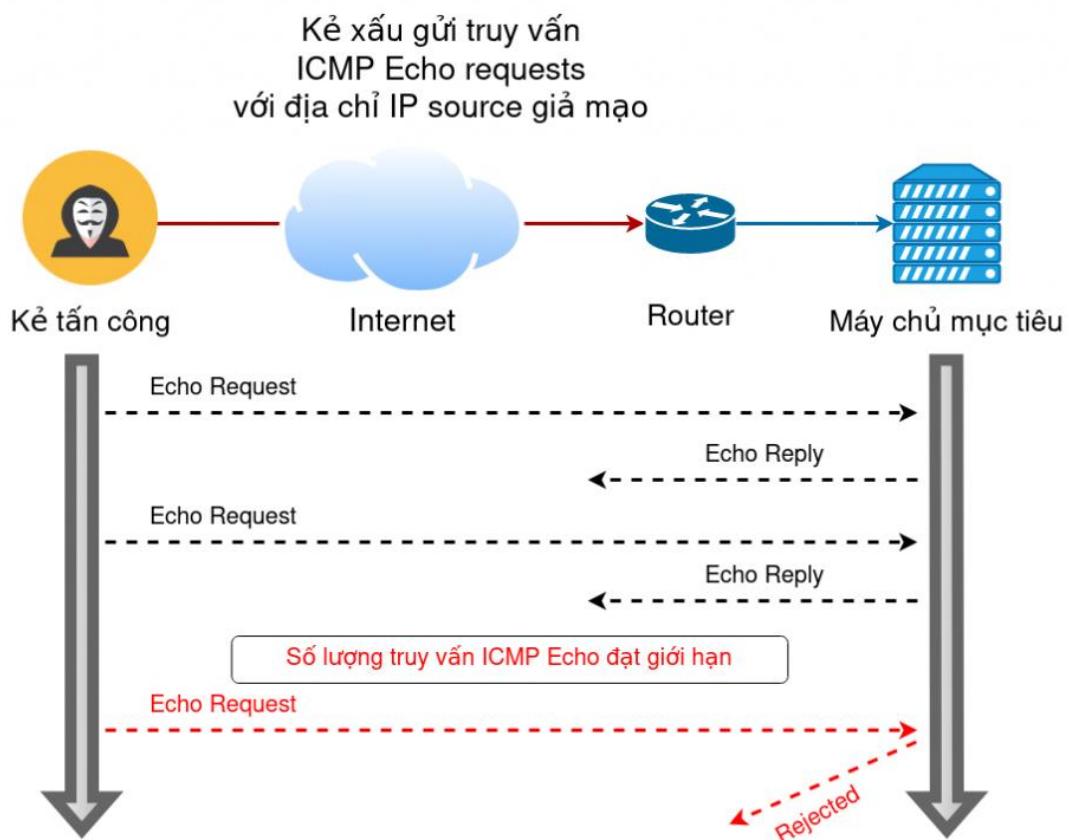
Hình 4.1.1 Luồng tấn công SYN Flood

#### 4.1.2 Tấn công ICMP Flood (Ping Flood)

ICMP Flood là kịch bản DoS trong đó hệ thống mục tiêu bị buộc phải xử lý số lượng lớn gói ICMP Echo Request trong thời gian ngắn

Kết quả:

- Băng thông mạng bị chiếm dụng
- CPU hệ thống tăng cao
- Người dùng hợp lệ không thể truy cập dịch vụ



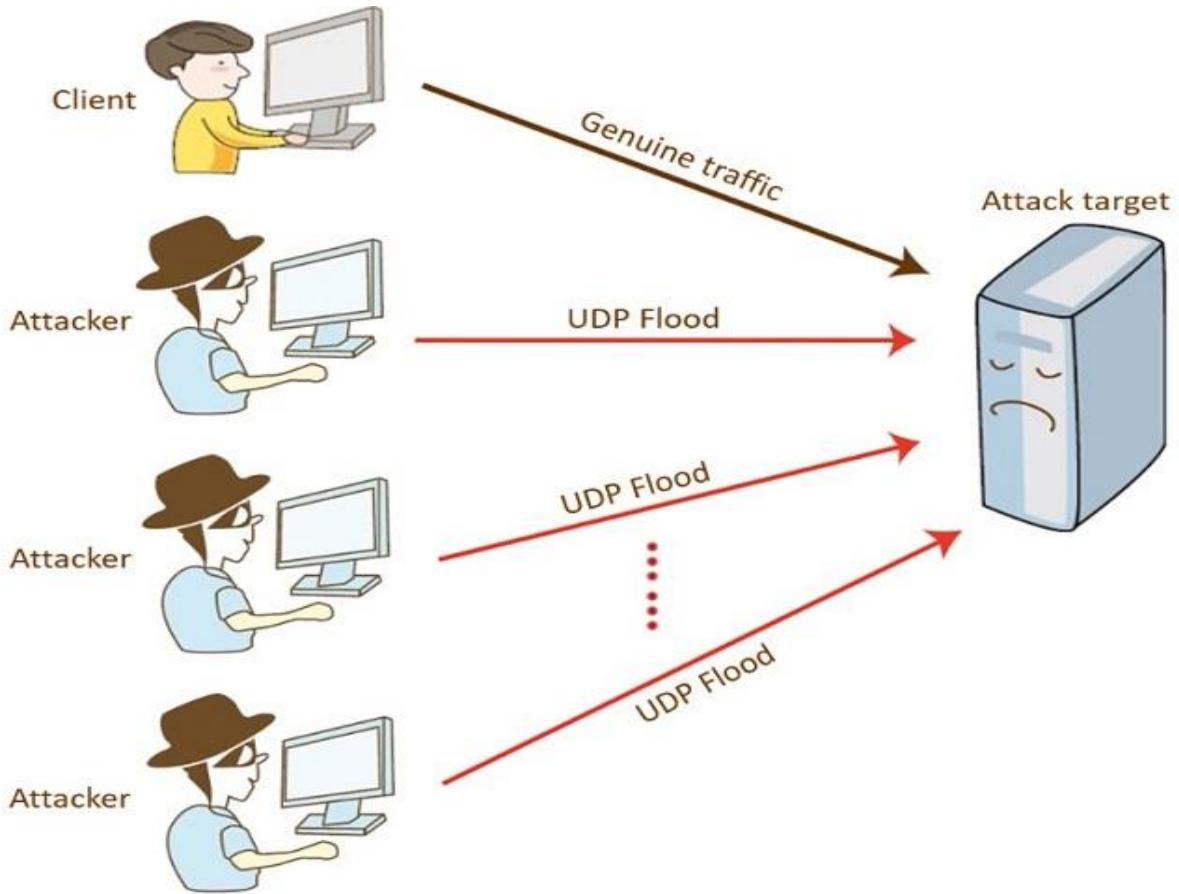
*Hình 4.1.2 Mô hình tấn công ICMP Flood*

#### 4.1.3 Tấn công UDP Flood

UDP Flood xảy ra khi kẻ tấn công gửi hàng loạt gói UDP đến nhiều cổng khác nhau trên hệ thống mục tiêu

Kết quả:

- Hệ thống phải xử lý lượng lớn gói tin không cần thiết
- CPU và băng thông bị quá tải
- Dịch vụ mạng bị gián đoạn



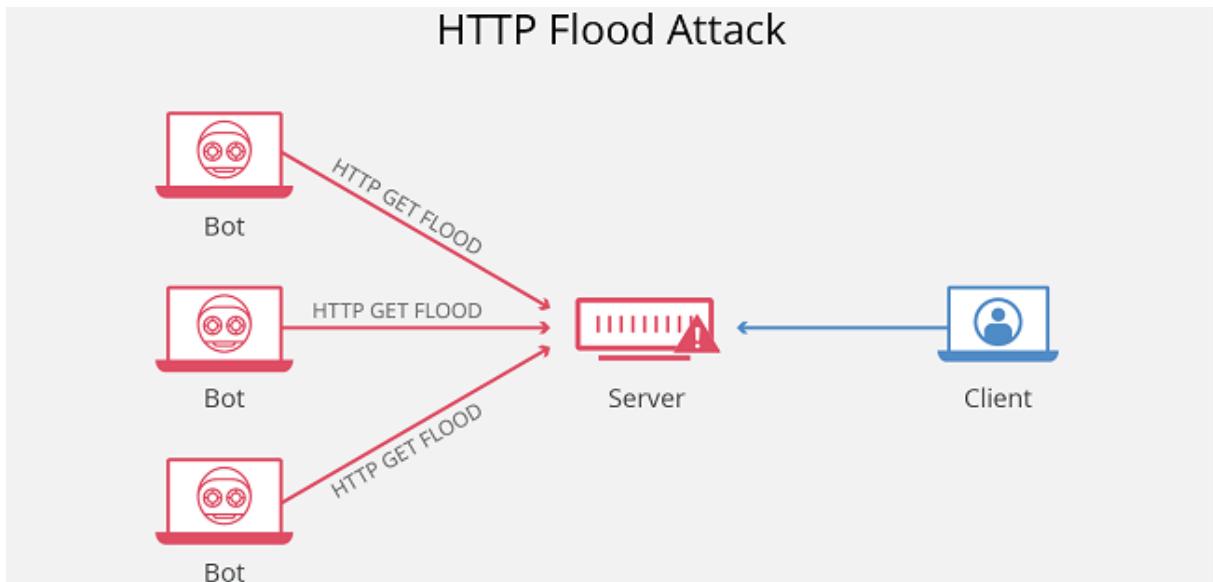
Hình 4.1.3 Tấn công UDP Flood

#### 4.1.4 Tấn công HTTP Flood

HTTP Flood là hình thức tấn công DoS ở tầng ứng dụng, trong đó kẻ tấn công gửi số lượng lớn các yêu cầu HTTP hợp lệ đến máy chủ web

Kết quả:

- CPU, RAM và cơ sở dữ liệu bị tiêu hao
- Website phản hồi chậm hoặc không phản hồi



Hình 4.1.4 Luồng tấn công HTTP Flood

## 4.2 Cách thức tấn công DDoS

DDoS (Distributed Denial of Service) là hình thức tấn công phân tán, trong đó nhiều thiết bị khác nhau đồng thời gửi lưu lượng đến hệ thống mục tiêu. Các thiết bị này thường bị điều khiển thông qua mô hình botnet

## 4.3 Mô hình botnet tấn công trong DDoS

### 4.3.1 Tổng quan

- Botnet là một mạng lưới các thiết bị bị kiểm soát trái phép, bao gồm máy tính, máy chủ và đặc biệt là các thiết bị IoT. Các thiết bị này bị nhiễm mã độc và hoạt động dưới sự điều khiển của kẻ tấn công thông qua máy chủ điều khiển trung tâm (Command & Control – C&C)

- Botnet đóng vai trò nền tảng cho các cuộc tấn công DDoS hiện đại nhờ khả năng phân tán nguồn tấn công, che giấu danh tính và mở rộng quy mô linh hoạt [4.1]

### 4.3.2 Các thành phần trong mô hình botnet

Một mô hình botnet điển hình bao gồm:

- Attacker: Khởi tạo và điều phối tấn công
- C&C Server: Trung gian phát lệnh, điều phối bot
- Bot/Zombie: Thiết bị bị nhiễm mã độc, thực thi lệnh
- Target: Hệ thống mục tiêu bị tấn công [4.2]

### 4.3.3 Quy trình hình thành botnet (Giai đoạn chuẩn bị)

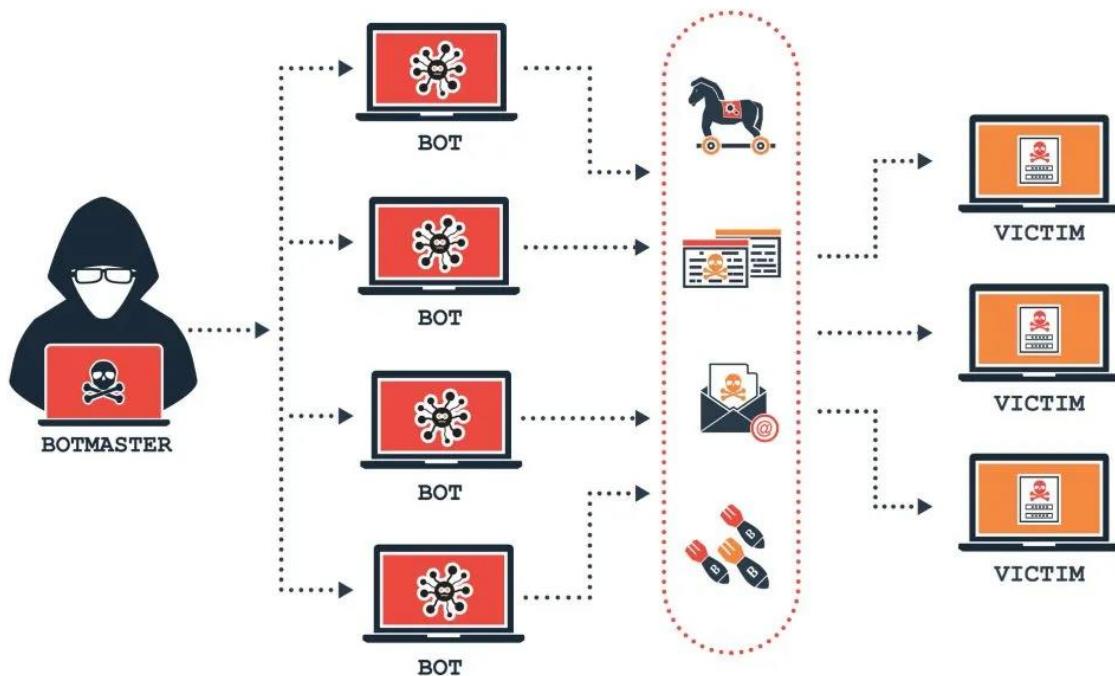
Quá trình xây dựng botnet thường bao gồm các bước:

1. Phát tán mã độc thông qua lỗ hổng, mật khẩu yếu hoặc phishing
2. Thiết bị nhiễm mã độc và kết nối về C&C
3. Botnet tự động mở rộng quy mô bằng cách quét và lây lan [4.3]

#### 4.3.4 Quy trình tấn công DDoS sử dụng botnet

Một cuộc tấn công DDoS dựa trên botnet thường diễn ra theo các bước sau:

1. Attacker xác định mục tiêu
2. Lựa chọn chiến thuật và loại tấn công
3. Phát lệnh qua C&C Server
4. Các bot đồng loạt gửi lưu lượng tấn công
5. Hệ thống mục tiêu bị quá tải và gián đoạn dịch vụ [4.4]



Hình 4.3.2 Quy trình tổng quát tấn công DDoS bằng botnet

#### 4.3.5 Đặc điểm nguy hiểm của mô hình botnet

Mô hình botnet làm cho tấn công DDoS trở nên đặc biệt nguy hiểm do:

- Tính phân tán cao
- Khó truy vết nguồn gốc
- Dễ mở rộng quy mô

- Có thể triển khai tấn công đa vector [4.5]

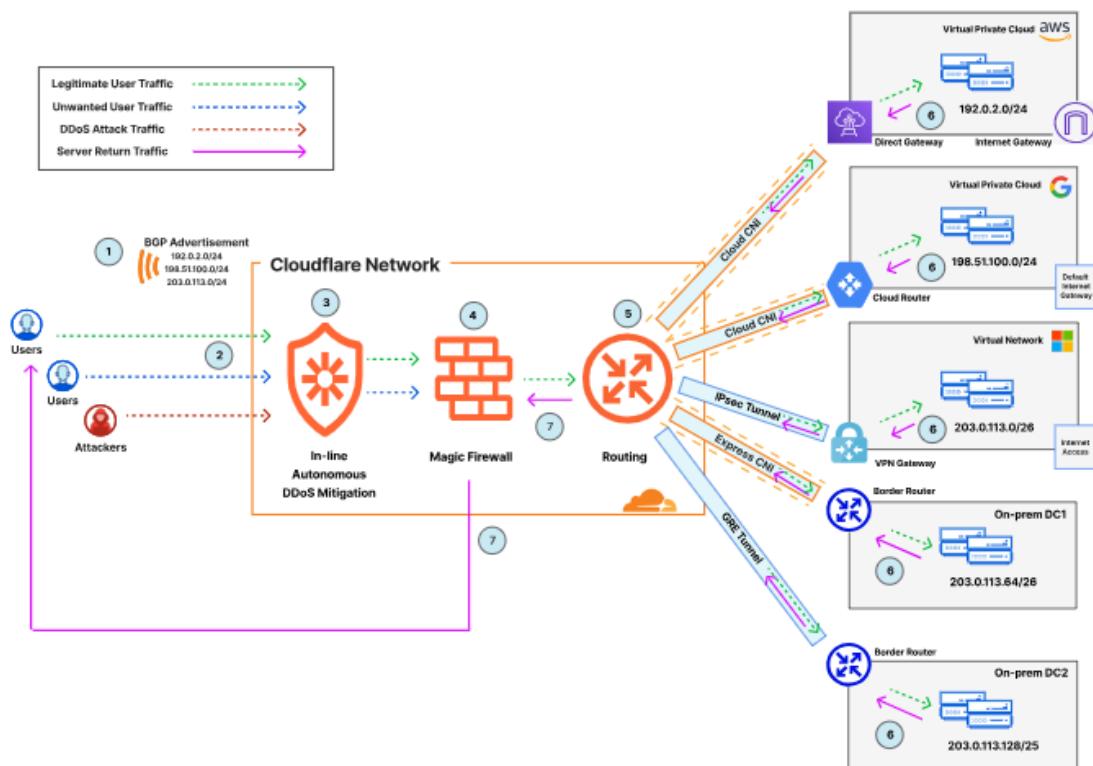
#### 4.4 Nhận xét và đánh giá

- Qua việc phân tích mô hình botnet và quy trình tấn công DDoS, có thể thấy rằng botnet chính là nền tảng cốt lõi của các cuộc tấn công DDoS hiện đại. Mức độ nguy hiểm của DDoS không chỉ đến từ từng kỹ thuật tấn công riêng lẻ mà chủ yếu xuất phát từ khả năng điều phối tập trung và quy mô lớn của mạng botnet
- Nội dung chương này là cơ sở để xây dựng và đánh giá các biện pháp phòng chống DDoS hiệu quả, sẽ được trình bày trong Chương 5

## CHƯƠNG 5 - CÁC BIỆN PHÁP PHÒNG CHỐNG TẤN CÔNG DOS/DDOS

### 5.1 Phòng chống ở mức mạng (Network Level)

Đây là tuyến phòng thủ đầu tiên, tập trung vào việc kiểm soát luồng lưu lượng đi vào hệ thống mạng (qua Routers, Switches, Firewalls) để chặn các cuộc tấn công chiếm dụng băng thông (Volumetric Attacks)



Hình 5.1: Sơ đồ chặn lọc gói tin tại Firewall

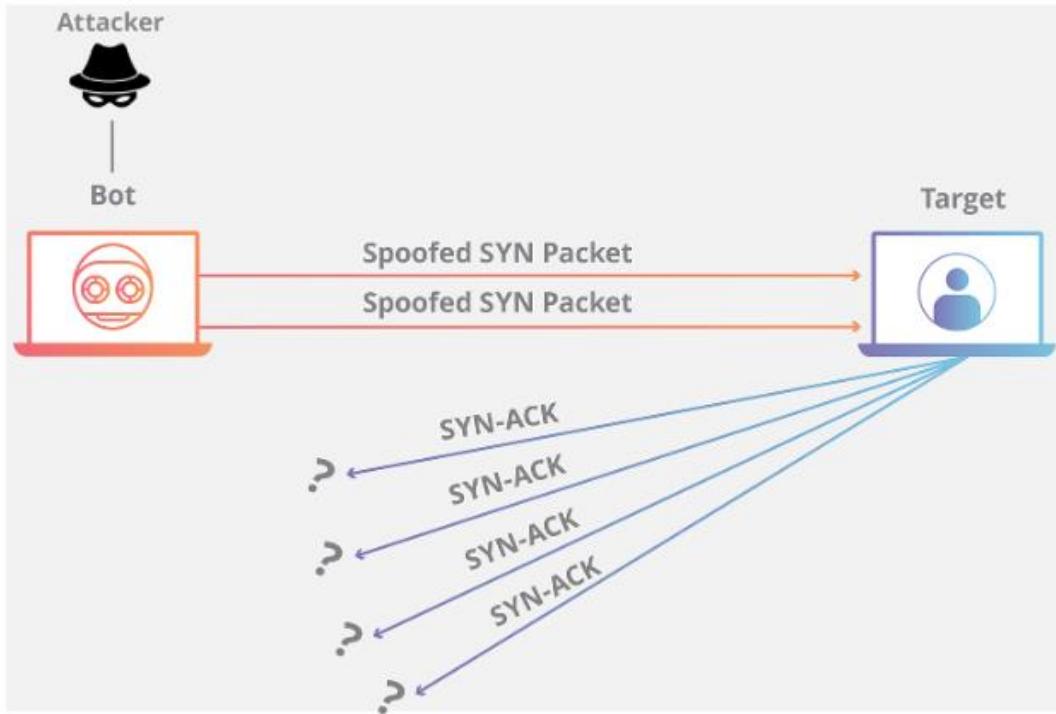
- Mô tả: Hình ảnh minh họa một thiết bị Firewall (Tường lửa) đứng trước hệ thống mạng nội bộ, thực hiện lọc các gói tin độc hại dựa trên địa chỉ IP hoặc cổng (Port) trước khi chúng đến được Server

Các giải pháp cụ thể:

- Ingress/Egress Filtering (Lọc đầu vào/ra):
  - Áp dụng chuẩn BCP 38 để chặn các gói tin đi vào mạng có địa chỉ IP nguồn giả mạo (IP Spoofing). Đồng thời, đảm bảo không có gói tin giả mạo nào đi ra từ mạng nội bộ, ngăn chặn hệ thống trở thành một phần của Botnet [5.2]
- Danh sách kiểm soát truy cập (Access Control Lists - ACLs):
  - Cấu hình Router/Firewall để chặn (Deny) các địa chỉ IP nguồn đáng ngờ hoặc các dải IP không thuộc phạm vi cung cấp dịch vụ
  - Chặn các giao thức không cần thiết (Ví dụ: chặn UDP nếu chỉ chạy Web Server) để giảm thiểu bê mặt tấn công [5.3]
- Rate Limiting (Giới hạn tốc độ):
  - Thiết lập giới hạn số lượng gói tin (PPS) hoặc băng thông (Mbps) tối đa từ một địa chỉ IP cụ thể trong một đơn vị thời gian. Điều này ngăn chặn một thiết bị đơn lẻ làm tràn ngập mạng
- Blackholing / Sinkholing:
  - Trong trường hợp tấn công quá lớn, định tuyến toàn bộ lưu lượng đến một "hố đen" (Null route) để loại bỏ, chấp nhận hi sinh truy cập để bảo vệ hạ tầng cốt lõi [5.4]

## 5.2 Phòng chống ở mức hệ điều hành (Operating System Level)

Biện pháp này tập trung vào việc tinh chỉnh (hardening) cấu hình Kernel và ngăn xếp giao thức TCP/IP của máy chủ để chống lại các cuộc tấn công cạn kiệt tài nguyên (Protocol Attacks), điển hình là SYN Flood [5.5]



Hình 5.2: Cơ chế tấn công SYN Flood và cách phòng chống

- Mô tả: Hình ảnh so sánh giữa quy trình bắt tay 3 bước TCP bình thường (Normal Handshake) và khi bị tấn công SYN Flood (kẻ tấn công gửi nhiều gói SYN nhưng không gửi ACK, làm đầy bộ nhớ Server)

Các giải pháp cụ thể:

- Bật tính năng SYN Cookies:
  - Theo chuẩn RFC 4987, khi hàng đợi kết nối bị đầy, Server sẽ gửi lại một mã hóa (cookie) trong gói SYN-ACK mà không cấp phát bộ nhớ ngay lập tức. Bộ nhớ chỉ được cấp khi nhận được gói ACK cuối cùng hợp lệ từ Client [5.5]
- Tối ưu hóa các tham số TCP (TCP Tuning):
  - Tăng kích thước hàng đợi (Backlog Queue): Tăng giá trị net.ipv4.tcp\_max\_syn\_backlog (trên Linux) để hệ thống có thể chấp nhận nhiều kết nối chờ hơn
  - Giảm thời gian Timeout: Giảm thời gian chờ (tcp\_fin\_timeout) để hệ thống nhanh chóng đóng các kết nối "treo", thu hồi tài nguyên RAM/CPU [5.6]
- Hạn chế kết nối (Connection Limits):
  - Sử dụng iptables hoặc cấu hình hệ điều hành để giới hạn số lượng kết nối đồng thời tối đa cho phép từ một địa chỉ IP

### 5.3 Phòng chống ở mức ứng dụng (Application Level)

Tầng ứng dụng (Layer 7) là mục tiêu của các cuộc tấn công tinh vi (như HTTP Flood, Slowloris), thường có lưu lượng thấp nhưng nhắm vào logic xử lý nặng nề của ứng dụng để gây quá tải CPU/RAM



Hình 5.3: Mô hình Web Application Firewall (WAF)

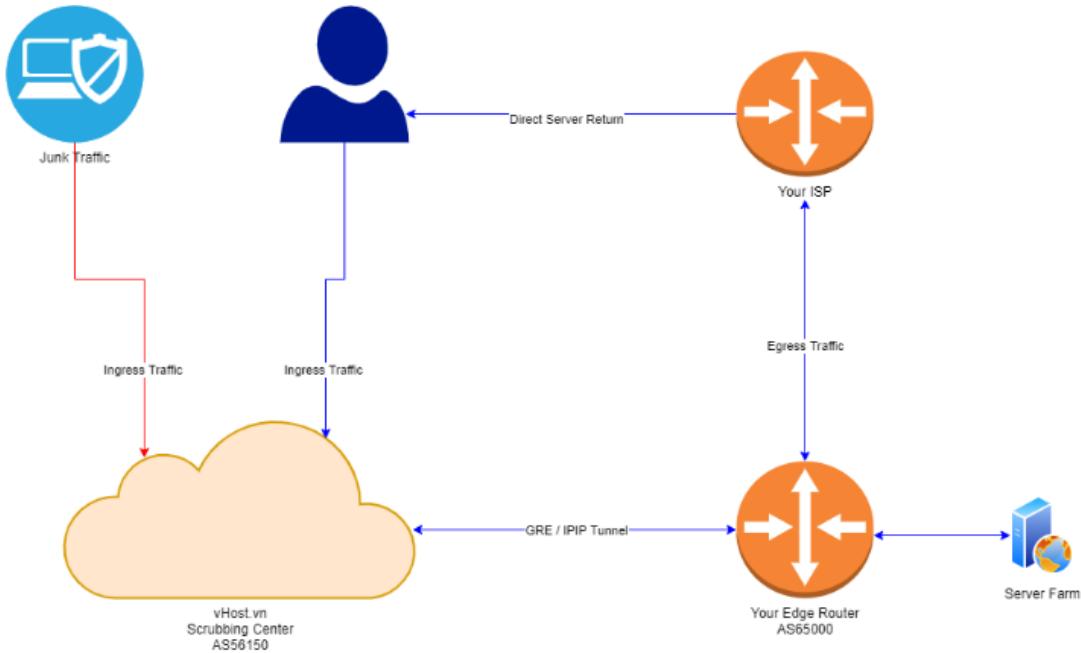
- Mô tả: Sơ đồ hiển thị WAF đứng trước Web Server. WAF đóng vai trò như một lá chắn, phân tích nội dung HTTP request để chặn các request độc hại trước khi chúng chạm tới ứng dụng

Các giải pháp cụ thể:

- Sử dụng Web Application Firewall (WAF):
  - Triển khai WAF (dựa trên các bộ luật như OWASP Core Rule Set) để phân tích nội dung gói tin HTTP/HTTPS. WAF giúp phát hiện các dấu hiệu bất thường như User-Agent lạ, SQL Injection, hoặc hành vi request lặp lại đáng ngờ [5.7]
- Cơ chế xác thực con người (Challenge-Response):
  - Sử dụng CAPTCHA (như reCAPTCHA) hoặc JavaScript Challenge tại các điểm nhạy cảm (trang đăng nhập, ô tìm kiếm) để phân biệt giữa người dùng thật và Bot tự động [5.8]
- Tối ưu hóa bộ nhớ đệm (Caching):
  - Sử dụng các giải pháp Cache (Redis, Varnish) hoặc CDN để phục vụ các nội dung tĩnh. Việc này giúp giảm tải đáng kể cho Database và Server gốc [5.9]

### 5.4 Phòng chống bằng dịch vụ và giải pháp chuyên dụng

Đối với các cuộc tấn công DDoS quy mô lớn (vài chục đến vài trăm Gbps), hạ tầng tự có của doanh nghiệp không thể chịu đựng nổi. Cần sử dụng các giải pháp Cloud-based hoặc ISP-based



*Hình 5.4: Kiến trúc Scrubbing Center (Trung tâm làm sạch)*

- Mô tả: Sơ đồ hiển thị quy trình: Khi có tấn công, lưu lượng được chuyển hướng (Redirect) qua Scrubbing Center. Tại đây, lưu lượng xấu bị lọc bỏ, chỉ có lưu lượng sạch (Clean traffic) được trả về Server gốc

Các giải pháp cụ thể:

- Mạng phân phối nội dung (CDN):
  - Sử dụng CDN để phân tán lưu lượng truy cập ra hàng nghìn máy chủ biên (Edge Server) trên toàn cầu, giúp hấp thụ lưu lượng tấn công thay vì dồn vào một điểm duy nhất [5.10]
- Scrubbing Centers (Trung tâm làm sạch lưu lượng):
  - Lưu lượng mạng được chuyển hướng (qua thay đổi DNS hoặc quảng bá BGP) đến các trung tâm xử lý chuyên dụng. Các trung tâm này có băng thông cực lớn để "hứng" đòn tấn công và lọc sạch dữ liệu [5.11]
- Anycast Routing:
  - Kỹ thuật định tuyến giúp quảng bá cùng một địa chỉ IP từ nhiều vị trí địa lý khác nhau. Khi bị tấn công, lưu lượng sẽ bị phân tán đến các node mạng gần nhất, ngăn chặn việc tấn công tập trung

## 5.5. So sánh các biện pháp phòng chống

Dưới đây là bảng tổng hợp ưu nhược điểm để lựa chọn giải pháp phù hợp:

Tiêu chí	Mức mạng (Network)	Mức hệ điều hành (OS)	Mức ứng dụng (App)	Dịch vụ chuyên dụng (Cloud)
<b>Mục tiêu chính</b>	Chặn tấn công băng thông lớn, lọc IP	Chặn tấn công cạn kiệt tài nguyên (TCP state)	Chặn tấn công vào logic, lỗi ứng dụng (Layer 7).	Chặn mọi loại tấn công quy mô cực lớn
<b>Công cụ điển hình</b>	Firewall, Router ACL, IPS	Kernel tuning, IPTables	WAF, CAPTCHA, Code optimization	Scrubbing Centers, CDN, Anycast
<b>Ưu điểm</b>	Hiệu quả chặn sớm. Chi phí thấp nếu dùng thiết bị có sẵn	Tăng khả năng chịu tải mà không tốn chi phí phần cứng	Phân biệt chính xác Bot và người dùng thật	Khả năng chịu tải gần như vô hạn. Dễ triển khai
<b>Nhược điểm</b>	Khó chặn được tấn công Layer 7. Dễ nghẽn đường truyền	Giới hạn bởi phần cứng vật lý (CPU/RAM)	Có thể ảnh hưởng trải nghiệm người dùng (CAPTCHA)	Chi phí cao. Phụ thuộc bên thứ 3
<b>Hiệu quả với DDoS lớn</b>	Thấp	Thấp	Thấp	Rất cao

## CHƯƠNG 6 - THỰC HÀNH, MÔ PHỎNG VÀ ĐÁNH GIÁ

### 6.1 Mục tiêu thí nghiệm

- Tìm hiểu cách thức hoạt động của tấn công DoS/DDoS ở tầng mạng và tầng ứng dụng
- Thực hiện thực nghiệm HTTP Flood (Layer 7) trong môi trường ảo hóa
- Phân tích tác động của tấn công lên hệ thống
- Triển khai và đánh giá biện pháp phòng thủ bằng Nginx rate limiting
- So sánh trước và sau khi phòng thủ

### 6.2 Mô hình và môi trường thí nghiệm

**6.2.1 Mô hình mạng:** Môi trường thí nghiệm được triển khai trên nền tảng ảo hóa VMware với 02 máy ảo, trong đó các vai trò logic được phân tách thông qua nhiều phiên terminal, phản ánh đúng quá trình thực hành tấn công và phòng chống DoS/DDoS tầng ứng dụng (Layer 7)

### 6.2.1.1 Kali Linux (Attacker & Client mô phỏng)

- Terminal 1 - Điều khiển kịch bản: Dùng để khởi động, dừng tấn công và theo dõi kết quả đầu ra của các công cụ thử nghiệm
- Terminal 2 - Client hợp lệ: Mô phỏng người dùng hợp lệ, đo thời gian phản hồi và kiểm tra khả năng truy cập dịch vụ bằng curl
- Terminal 3 - HTTP Flood Generator: Thực hiện tấn công HTTP Flood bằng ApacheBench (ab) và các công cụ tạo lưu lượng

**6.2.1.2 Ubuntu Server (Victim):** Ubuntu Server không phân tách nhiều terminal chuyên biệt, mà sử dụng console/terminal duy nhất để quản lý dịch vụ Nginx, theo dõi CPU, RAM trong suốt quá trình tấn công và phòng thủ

- Console / Terminal – Web Server & giám sát tài nguyên:

- + Vận hành dịch vụ Nginx
- + Giám sát tài nguyên hệ thống bằng htop

### 6.2.1.3 Sơ đồ mô hình thí nghiệm

└─ Terminal A: HTTP Flood (ApacheBench)

└─ Terminal B: Client hợp lệ (curl)

└─ Terminal C: Điều khiển kịch bản



└─ (VMware NAT)



Ubuntu Server (Nginx)

└─ Console: Vận hành Nginx + giám sát (htop)

→ Trong quá trình thực hành, Ubuntu Server được quản lý thông qua console duy nhất để vận hành dịch vụ Nginx và giám sát tài nguyên hệ thống, trong khi các vai trò tấn công và mô phỏng client được phân tách thông qua nhiều phiên terminal trên Kali Linux

Kết nối mạng: Các máy ảo được kết nối thông qua mạng NAT của VMware, đảm bảo môi trường thí nghiệm khép kín và an toàn

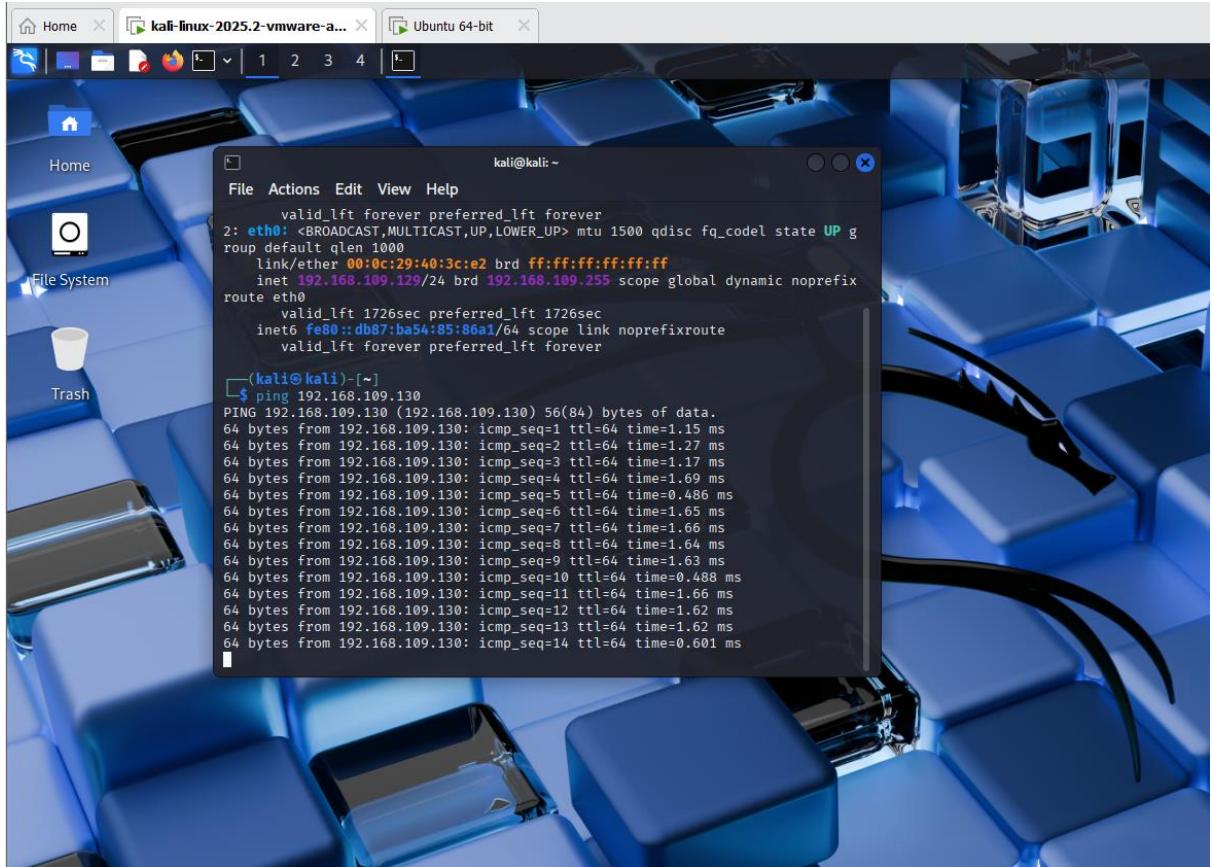
### 6.2.2 Công cụ sử dụng

Thành phần	Công cụ
Web server	Nginx
Sinh lưu lượng tấn công	ApacheBench (ab), hping3
Quan sát tài nguyên	htop
Phân tích truy cập	access.log (Nginx)

## 6.3 Chuẩn bị môi trường

### 6.3.1 Kiểm tra kết nối mạng

- Từ Kali ping sang Ubuntu:



- Từ Ubuntu ping ngược lại Kali:

```

victim-server login: ladadmin
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-90-generic x86_64)

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

System information as of Mon Dec 22 07:10:37 PM UTC 2025

System load: 1.08 Processes: 250
Usage of /: 26.3% of 9.75GB Users logged in: 0
Memory usage: 7% IPv4 address for ens33: 192.168.109.130
Swap usage: 0%

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

ladadmin@victim-server:~$ 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
    link/ether 00:0c:29:83:a2:87 brd ff:ff:ff:ff:ff:ff
        inet 192.168.109.130/24 brd 192.168.109.255 scope global dynamic ens33
            valid_lft 1754sec preferred_lft 1754sec
            inet6 fe80::20c:29ff:fe83:a287/64 scope link
                valid_lft forever preferred_lft forever
ladadmin@victim-server:~$ ping 192.168.109.129
PING 192.168.109.129 (192.168.109.129) 56(84) bytes of data.
64 bytes from 192.168.109.129: icmp_seq=1 ttl=64 time=0.609 ms
64 bytes from 192.168.109.129: icmp_seq=2 ttl=64 time=1.79 ms
64 bytes from 192.168.109.129: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.109.129: icmp_seq=4 ttl=64 time=1.69 ms
64 bytes from 192.168.109.129: icmp_seq=5 ttl=64 time=0.517 ms
64 bytes from 192.168.109.129: icmp_seq=6 ttl=64 time=0.626 ms
64 bytes from 192.168.109.129: icmp_seq=7 ttl=64 time=1.63 ms
64 bytes from 192.168.109.129: icmp_seq=8 ttl=64 time=0.564 ms
64 bytes from 192.168.109.129: icmp_seq=9 ttl=64 time=1.56 ms

```

→ Ping thành công, xác nhận hai máy thông nhau

### 6.3.2 Cài đặt và kiểm tra Nginx (Ubuntu)

- Cài đặt và kiểm tra trạng thái

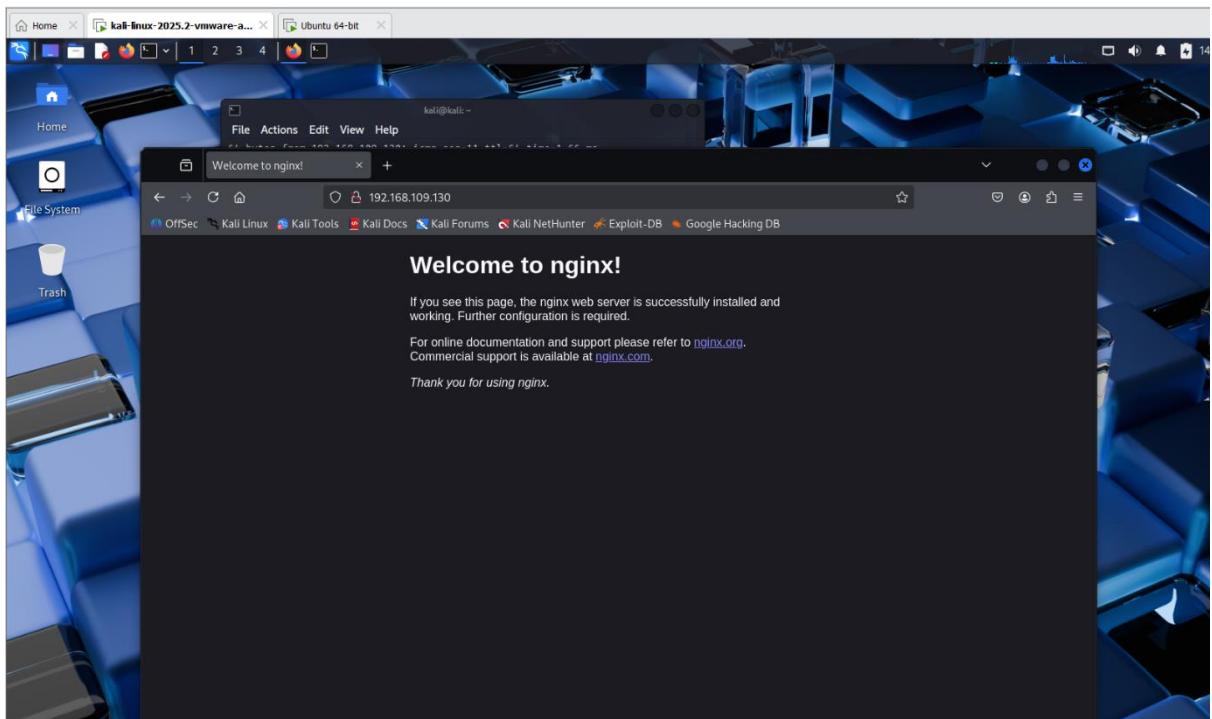
```

192.168.109.129      curl 1           http://192.168.109.130  ip          sudo
ladadmin@victim-server:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-12-22 19:10:25 UTC; 4min 20s ago
       Docs: man/nginx(8)
   Process: 1174 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 1158 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 1177 (nginx)
   Tasks: 3 (limit: 4545)
  Memory: 3.7M (peak: 3.9M)
    CPU: 44ms
   CGroup: /system.slice/nginx.service
           └─1177 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─1178 "nginx: worker process"
             ├─1179 "nginx: worker process"
             └─1180 "nginx: worker process"

Dec 22 19:10:24 victim-server systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Dec 22 19:10:25 victim-server systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ladadmin@victim-server:~$ _

```

- Truy cập từ kali



→ Web hiển thị bình thường

### 6.3.3 Đo baseline (trước tấn công)

- Mở giám sát CPU: htop

```

[1] 0: [Main]
[1] 1: [I/O]
[1] 2: [Sup]
[1] 3: [Mem]

Tasks: 26, 25 thr, 189 kthr; 1 running
0.0% Load average: 0.00 0.04 0.04
238M/3.76G Uptime: 0:00:45

PID USER PRI NI VIRT RES SHR S CPU%MEM% TIME+ Command
1 root 20 0 21956 12760 5924 S 0.0 0.3 0:00:34 /sbin/init
459 root 19 -1 66932 10834 17880 S 0.0 0.5 0:00:26 /usr/lib/systemd/systemd-journald
502 root RT 0 30524 20704 1064 S 0.0 0.0 0:00:00 /sbin/multipathd -d -s
511 root 20 0 30524 3368 1064 S 0.0 0.2 0:00:30 /usr/lib/systemd/systemd-udevd
517 root 20 0 346M 27392 8704 S 0.0 0.7 0:00:00 /sbin/multipathd -d -s
518 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:00 /sbin/multipathd -d -s
519 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:00 /sbin/multipathd -d -s
520 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:00 /sbin/multipathd -d -s
521 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:00 /sbin/multipathd -d -s
522 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:00 /sbin/multipathd -d -s
636 systemd-ne 20 0 19088 9472 8328 S 0.0 0.2 0:00:04 /usr/lib/systemd/systemd-networkd
654 systemd-re 20 0 21580 12672 10496 S 0.0 0.3 0:00:11 /usr/lib/systemd/systemd-resolved
669 systemd-ti 20 0 31024 7656 6784 S 0.0 0.2 0:00:05 /usr/lib/systemd/systemd-timesyncd
713 systemd-udevd 20 0 33644 12160 1024 S 0.0 0.3 0:00:06 /usr/lib/systemd/systemd-udevd
754 root 20 0 236M 3216 7936 S 0.0 0.3 0:01:30 /usr/bin/vmtoolsd
802 messagebus 20 0 9732 5376 4608 S 0.0 0.1 0:00:06 @dbus-daemon --system -address=@systemd: --nofork --nopidfile --systemd-activation --syslog-o
821 root 20 0 236M 3216 7936 S 0.0 0.2 0:00:00 /usr/bin/vmtoolsd
825 polkitd 20 0 3008 7808 7648 S 0.0 0.2 0:00:04 /usr/lib/polkit-1/polkitd --no-debug
841 root 20 0 236M 3216 7936 S 0.0 0.2 0:00:00 /usr/bin/vmtoolsd
842 root 20 0 236M 3216 7936 S 0.0 0.2 0:00:00 /usr/bin/vmtoolsd
849 root 20 0 18124 1832 7608 S 0.0 0.2 0:00:06 /usr/lib/systemd/systemd-logind
853 root 20 0 457M 11568 11932 S 0.0 0.3 0:00:08 /usr/libexec/udisks2/udisksd
921 root 20 0 457M 11568 11932 S 0.0 0.3 0:00:01 /usr/libexec/udisks2/udisksd
923 root 20 0 3008 7808 7648 S 0.0 0.2 0:00:00 /usr/libexec/udisks2/udisksd
928 root 20 0 3008 7808 7648 S 0.0 0.3 0:00:00 /usr/libexec/udisks2/udisksd
956 root 20 0 167M 22912 13568 S 0.0 0.6 0:00:21 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
963 syslog 20 0 217M 6016 4488 S 0.0 0.2 0:00:03 /usr/sbin/syslogd -n -INONE
968 polkitd 20 0 3008 7808 7648 S 0.0 0.2 0:00:00 /usr/lib/polkit-1/polkitd --no-debug
969 polkitd 20 0 3008 7808 7648 S 0.0 0.2 0:00:00 /usr/lib/polkit-1/polkitd --no-debug
970 polkitd 20 0 3008 7808 7648 S 0.0 0.2 0:00:00 /usr/lib/polkit-1/polkitd --no-debug
991 root 20 0 3008 11568 11932 S 0.0 0.3 0:00:00 /usr/libexec/udisks2/udisksd
992 root 20 0 302M 12328 10880 S 0.0 0.3 0:00:10 /usr/sbin/modemManager
1004 root 20 0 457M 11568 11932 S 0.0 0.3 0:00:00 /usr/libexec/udisks2/udisksd
1010 syslog 20 0 217M 6016 4488 S 0.0 0.2 0:00:00 /usr/sbin/syslogd -n -INONE
1011 syslog 20 0 217M 6016 4488 S 0.0 0.2 0:00:00 /usr/sbin/syslogd -n -INONE
1012 syslog 20 0 217M 6016 4488 S 0.0 0.2 0:00:00 /usr/sbin/syslogd -n -INONE
1020 root 20 0 302M 12328 10880 S 0.0 0.3 0:00:00 /usr/sbin/modemManager
1023 root 20 0 302M 12328 10880 S 0.0 0.3 0:00:00 /usr/sbin/modemManager
1027 root 20 0 302M 12328 10880 S 0.0 0.3 0:00:00 /usr/sbin/modemManager
1115 root 20 0 6824 2688 2564 S 0.0 0.1 0:00:01 /usr/sbin/cron -f -P

F1 Help F2 Setup F3 Search F4 Filter F5 Free F6 Sort By F7 Nice F8 Kill F9 Built

```

- Quan sát ta thấy:

+ CPU thấp

+ RAM ổn định

+ Load average thấp

- Ghi nhận traffic mạng bình thường: sudo tcpdump -i <interface> -n

```

ladadmin@victim-server:~$ sudo tcpdump -i ens33 -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
19:23:47.164724 IP 192.168.109.129.37868 > 193.70.115.65.123: NTPv4, Client, length 48
19:23:47.192935 IP 193.70.115.65.123 > 192.168.109.129.37868: NTPv4, Server, length 48
19:23:52.192621 ARP, Request who-has 192.168.109.2 tell 192.168.109.129, length 46
19:23:52.192621 ARP, Reply 192.168.109.2 is-at 00:50:56:e1:6a:fe, length 46
19:24:14.902926 IP 192.168.109.129.51554 > 193.70.115.65.123: NTPv4, Client, length 48
19:24:14.928460 IP 193.70.115.65.123 > 192.168.109.129.51554: NTPv4, Server, length 48
19:24:15.265578 ARP, Request who-has 192.168.109.254 tell 192.168.109.129, length 46
19:24:15.265578 ARP, Reply 192.168.109.254 is-at 00:50:56:fb:94:87, length 46
19:24:15.265819 IP 192.168.109.129.68 > 192.168.109.254.67: BOOTP/DHCP, Request from 00:0c:29:40:3c:e2, length 282
19:24:15.265819 IP 192.168.109.254.67 > 192.168.109.129.68: BOOTP/DHCP, Reply, length 300
19:24:45.562097 ARP, Request who-has 192.168.109.2 tell 192.168.109.1, length 46
19:24:46.631091 ARP, Request who-has 192.168.109.2 tell 192.168.109.1, length 46
19:24:47.165193 IP 192.168.109.129.47340 > 193.70.115.65.123: NTPv4, Client, length 48
19:24:47.194762 IP 193.70.115.65.123 > 192.168.109.129.47340: NTPv4, Server, length 48
19:24:47.549543 ARP, Request who-has 192.168.109.2 tell 192.168.109.1, length 46
19:24:48.557133 ARP, Request who-has 192.168.109.2 tell 192.168.109.1, length 46
19:24:49.296377 IP 192.168.109.130.68 > 192.168.109.254.67: BOOTP/DHCP, Request from 00:0c:29:83:a2:87, length 298
19:24:49.297484 IP 192.168.109.254.67 > 192.168.109.130.68: BOOTP/DHCP, Reply, length 300
19:24:52.350799 ARP, Request who-has 192.168.109.2 tell 192.168.109.129, length 46
19:24:52.350799 ARP, Reply 192.168.109.2 is-at 00:50:56:e1:6a:fe, length 46
-
```

+ Ít gói tin

+ Không có burst bất thường

- Đo response time (baseline)

```

└─[kali㉿kali]─[~]
$ curl -o /dev/null -s -w "Time: ${time_total}s\n" http://192.168.109.130
Time: 0.001962s

```

→ Response time thấp, ổn định

## 6.4 Tiến hành tấn công SYN FLOOD

- Mục tiêu:

+ Quan sát tác động của SYN Flood lên server

+ Thu thập số liệu ở trạng thái During attack

+ Chưa bật bất kỳ cơ chế phòng thủ nào

- Terminal:

Ubuntu (Victim): mở 2 terminal

+ Ubuntu local T1 (VM Ubuntu): giám sát tài nguyên (htop)

Kiểm tra SSH trên Ubuntu

```

• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Active: active (running) since Mon 2025-12-22 19:10:25 UTC; 21min ago
  TriggeredBy: • ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
  Process: 1131 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 1185 (sshd)
    Tasks: 1 (limit: 4545)
   Memory: 2.2M (peak: 2.4M)
      CPU: 42ms
     CGroup: /system.slice/ssh.service
             └─1185 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 22 19:10:24 victim-server systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Dec 22 19:10:25 victim-server sshd[1185]: Server listening on 0.0.0.0 port 22.
Dec 22 19:10:25 victim-server sshd[1185]: Server listening on :: port 22.
Dec 22 19:10:25 victim-server systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
ladadmin@victim-server:~$ 

```

+ Ubuntu SSH T2 (Kali → SSH): quan sát gói tin (tcpdump)

→ Chạy: ssh ladadmin@192.168.109.130 → Nhập password → Đăng nhập thành công

Kali (Attacker): mở 1 terminal để tấn công SYN

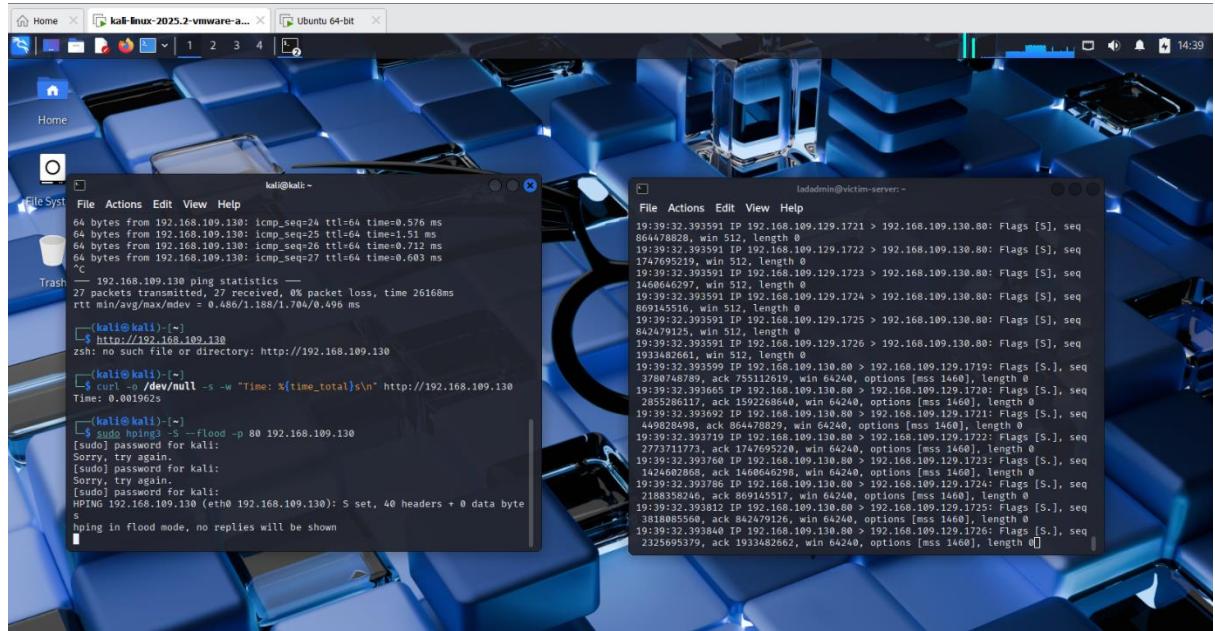
→ Lần này có chạy song song, khác với baseline

**6.4.1 Thực hiện tấn công:** sudo hping3 -S --flood -p 80 <IP\_UBUNTU>

- -S: SYN

- --flood: gửi liên tục (rất nhanh)

- -p 80: cung web



Quan sát trên terminal 1- htop và terminal 2 – tcpdump ta thấy:

- CPU / load tăng
  - Có thể thấy nhiều tiến trình/interrupt tăng
  - Rất nhiều gói SYN
  - Tần suất cao, liên tục

#### **6.4.2 Kết thúc tấn công**

Trên Kali: Ctrl + C

Trên Ubuntu:

System Status									
Process ID	User	PRI	NICE	VIRT	RES	SHR	S	CPU%+HEX%	TIME+
1799	ladadmin	20	0	6784	5120	3712	R	0.7 0.1	0:02.36 httpd
1851	tcpdump	20	0	26624	9856	8664	S	0.7 0.2	0:00.50 tcpdump -i ens33 -n tcp
1 root		20	0	21948	12984	9320	S	0.0 0.3	0:03.41 /sbin/init
450 root		19	-1	66832	18568	17936	S	0.0 0.5	0:00.29 /usr/lib/systemd/systemd-journald
502 root		RT	0	346K	27392	8764	S	0.0 0.7	0:00.18 /sbin/multipathd -d -s
511 root		20	0	30520	8568	4864	S	0.0 0.2	0:00.33 /usr/lib/systemd/systemd-udevd
517 root		20	0	346K	27392	8764	S	0.0 0.7	0:00.00 /sbin/multipathd -d -s
518 root		RT	0	346K	27392	8764	S	0.0 0.7	0:00.00 /sbin/multipathd -d -s
519 root		RT	0	346K	27392	8764	S	0.0 0.7	0:00.00 /sbin/multipathd -d -s
520 root		RT	0	346K	27392	8764	S	0.0 0.7	0:00.00 /sbin/multipathd -d -s
521 root		RT	0	346K	27392	8764	S	0.0 0.7	0:00.23 /sbin/multipathd -d -s
522 root		RT	0	346K	27392	8764	S	0.0 0.7	0:00.00 /sbin/multipathd -d -s
636 systemd-ne		20	0	15008	5472	8320	S	0.0 0.2	0:00.05 /usr/lib/systemd/systemd-networkd
654 systemd-re		20	0	21580	12672	10496	S	0.0 0.3	0:00.11 /usr/lib/systemd/systemd-resolved
660 systemd-ti		20	0	51024	7680	5784	S	0.0 0.2	0:00.05 /usr/lib/systemd/systemd-timesyncd
713 systemd-ti		20	0	51024	7680	5784	S	0.0 0.2	0:00.00 /usr/lib/systemd/systemd-timesyncd
754 root		20	0	53464	12168	10624	S	0.0 0.3	0:00.06 /usr/bin/vGauthservice
755 root		20	0	308K	5472	7936	S	0.0 0.2	0:00.40 /usr/bin/vmtoolsd
802 messagebus		20	0	9772	5376	4688	S	0.0 0.1	0:00.08 @ibus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-o
821 root		20	0	300K	5472	7936	S	0.0 0.2	0:00.00 /usr/bin/vmtoolsd
825 polkitd		20	0	300K	7040	7040	S	0.0 0.2	0:00.04 /usr/lib/polkit-1/polkitd --no-debug
841 root		20	0	308K	5472	7936	S	0.0 0.2	0:00.07 /usr/bin/vmtoolsd
842 root		20	0	308K	5472	7936	S	0.0 0.2	0:00.00 /usr/bin/vmtoolsd
849 root		20	0	18124	8832	7808	S	0.0 0.2	0:00.07 /usr/lib/systemd/systemd-logind
853 root		20	0	457M	13568	11392	S	0.0 0.3	0:00.09 /usr/libexec/udisks2/udisksd
921 root		20	0	457M	13568	11392	S	0.0 0.3	0:00.02 /usr/libexec/udisks2/udisksd
923 root		20	0	457M	13568	11392	S	0.0 0.3	0:00.00 /usr/libexec/udisks2/udisksd
928 root		20	0	457M	13568	11392	S	0.0 0.3	0:00.00 /usr/libexec/udisks2/udisksd
956 root		20	0	187M	22312	13568	S	0.0 0.6	0:00.21 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
963 syslog		20	0	217M	6016	4480	S	0.0 0.2	0:00.03 /usr/sbin/rsyslogd -r -NONE
968 polkitd		20	0	300K	7040	7040	S	0.0 0.2	0:00.00 /usr/lib/polkit-1/polkitd --no-debug
969 polkitd		20	0	300K	7008	7040	S	0.0 0.2	0:00.00 /usr/lib/polkit-1/polkitd --no-debug
973 polkitd		20	0	300K	7008	7040	S	0.0 0.2	0:00.00 /usr/lib/polkit-1/polkitd --no-debug
991 root		20	0	457M	13568	11392	S	0.0 0.3	0:00.00 /usr/libexec/udisks2/udisksd
992 root		20	0	382K	12328	10880	S	0.0 0.3	0:00.19 /usr/sbin/ModemManager
1004 root		20	0	457M	13568	11392	S	0.0 0.3	0:00.00 /usr/libexec/udisks2/udisksd
1010 syslog		20	0	217M	6016	4480	S	0.0 0.2	0:00.00 /usr/sbin/rsyslogd -r -NONE
1011 syslog		20	0	217M	6016	4480	S	0.0 0.2	0:00.00 /usr/sbin/rsyslogd -r -NONE
1012 syslog		20	0	217M	6016	4480	S	0.0 0.2	0:00.00 /usr/sbin/rsyslogd -r -NONE
1020 root		20	0	382K	12328	10880	S	0.0 0.3	0:00.00 /usr/sbin/ModemManager
1023 root		20	0	382K	12328	10880	S	0.0 0.3	0:00.00 /usr/sbin/ModemManager

→ CPU dần giảm (nhưng chưa có phòng thủ)

### 6.4.3 Nhận xét:

- SYN Flood tạo nhiều half-open connections
  - Tài nguyên tăng nhanh
  - Người dùng hợp lệ bị ảnh hưởng
  - Chưa có phòng thủ nên hệ thống dễ bị suy giảm

→ Trong thí nghiệm SYN Flood, mức sử dụng CPU không tăng đột biến do tấn công chủ yếu tác động vào bảng kết nối TCP thay vì xử lý ứng dụng. Tuy nhiên, phân tích gói tin cho thấy số lượng lớn gói SYN liên tục, gây nguy cơ cạn tài nguyên kết nối và ảnh hưởng đến người dùng hợp lệ.

### **6.5 Phòng thủ SYN FLOOD (Layer 4)**

- Mục tiêu:

- + Kích hoạt SYN Cookies (kernel) và giới hạn kết nối (iptables)

- + Chạy lại đúng kịch bản SYN Flood
- + So sánh trước và sau khi chạy
- không đổi tham số tấn công, để so sánh công bằng

### 6.5.1 Bật SYN Cookies (Ubuntu)

```
ladadmin@victim-server:~$ sudo sysctl -w net.ipv4.tcp_syncookies=1
[sudo] password for ladadmin:
net.ipv4.tcp_syncookies = 1
ladadmin@victim-server:~$ _
```

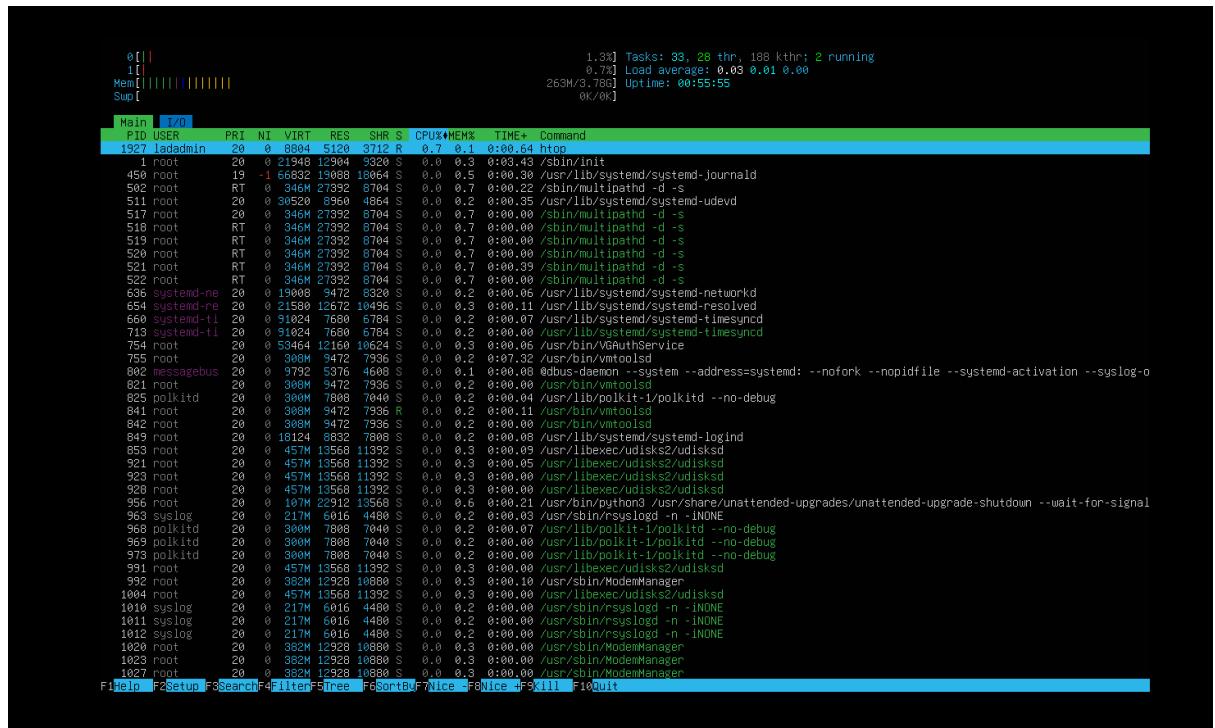
Note: SYN Cookies giảm half-open connections, hiệu quả với Layer 4

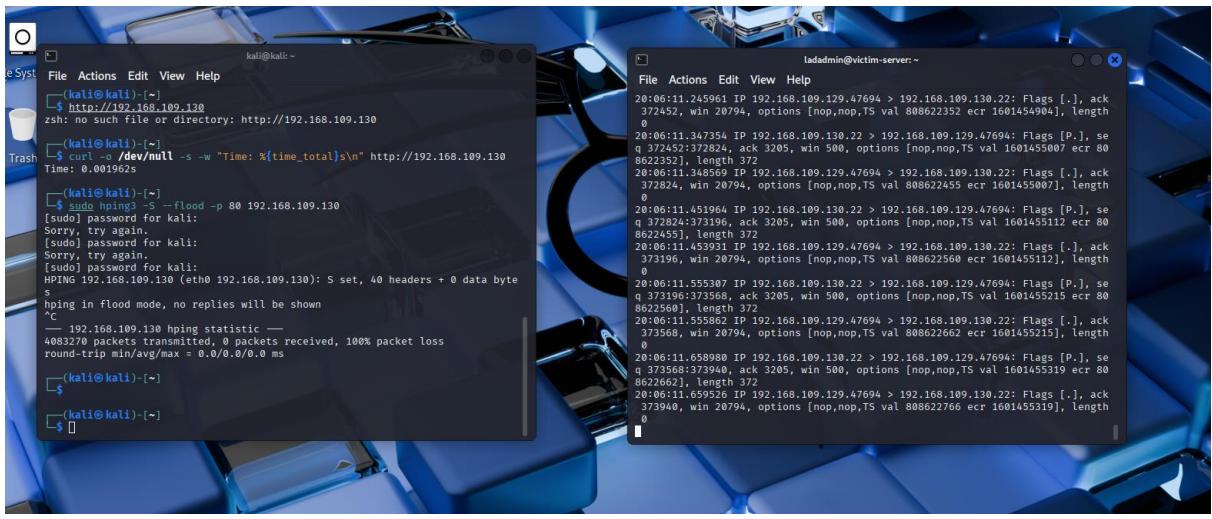
### 6.5.2 Thêm rule iptables giới hạn SYN (Ubuntu)

```
ladadmin@victim-server:~$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source          destination
1   ACCEPT     6    --  0.0.0.0/0        0.0.0.0/0
2   DROP       6    --  0.0.0.0/0        0.0.0.0/0      tcp dpt:80 flags:0x17/0x02 limit: avg 20/sec burst 40
Chain FORWARD (policy ACCEPT)
num  target     prot opt source          destination
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source          destination
ladadmin@victim-server:~$
```

Note: Có nguy cơ false positive nếu ngưỡng quá thấp

### 6.5.3 Chuẩn bị giám sát

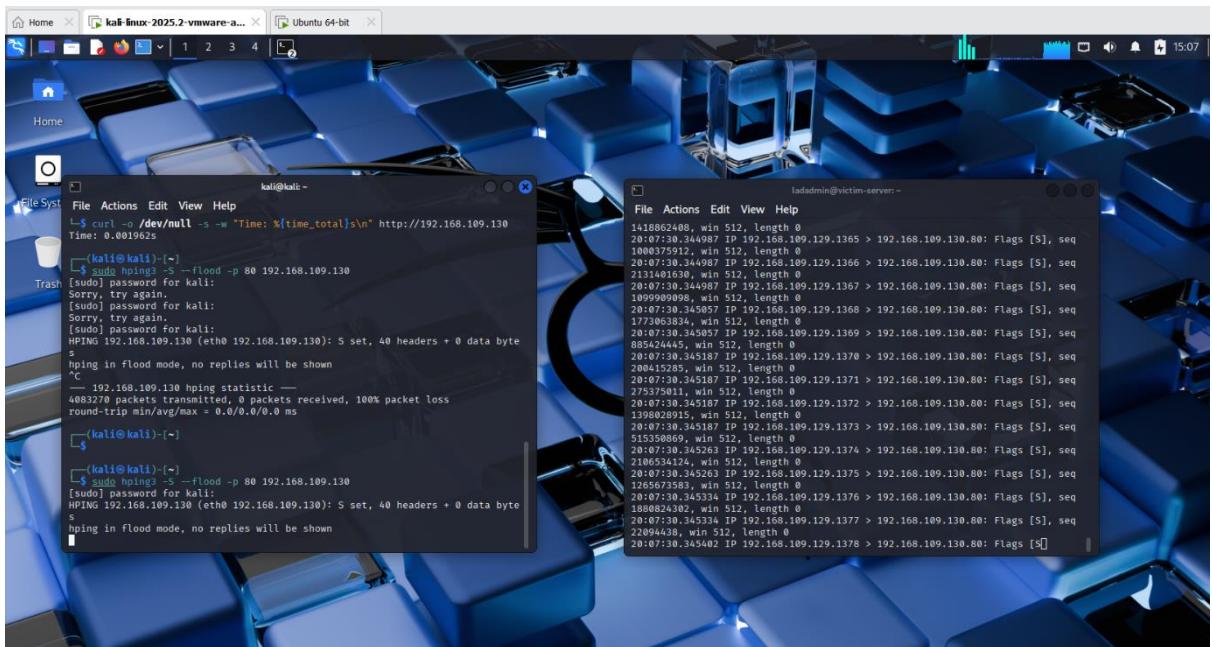




→ htop + tcpdump đang chạy (Before defense test)

#### 6.5.4 Chạy lại SYN Flood (Kali)

Trên Kali – terminal tấn công:



Main	I/O	PID	USER	PRI	NI	VIRT	RES	SHR	SI	CPU%	MEM%	TIME+	Command
0[]		755	root	20	0	308M	3472	7536	S	0.7	0.2	0:07:83	/usr/bin/vmtoolsd
1[		1783	ladadmin	20	0	16672	8912	4952	S	0.7	0.2	0:27.18	ssh: ladadmin@pts/0
Mem[		1	root	20	0	21548	12304	9328	S	0.0	0.3	0:03.44	/sbin/init
Sup[		450	root	19	-1	66832	19688	18644	S	0.0	0.5	0:00.30	/usr/lib/systemd/systemd-journald
		502	root	RT	0	346M	27392	8704	S	0.0	0.7	0:00.26	/sbin/multipathd -d -s
		511	root	20	0	30520	8956	4654	S	0.0	0.2	0:00.36	/usr/lib/systemd/systemd-udevd
		517	root	20	0	346M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
		518	root	RT	0	346M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
		519	root	RT	0	346M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
		520	root	RT	0	346M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
		521	root	RT	0	346M	27392	8704	S	0.0	0.7	0:00.41	/sbin/multipathd -d -s
		522	root	RT	0	346M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
		636	systemd-ne	20	0	15008	9472	8328	S	0.0	0.2	0:00.05	/usr/lib/systemd/systemd-networkd
		654	systemd-re	20	0	21580	12672	10496	S	0.0	0.3	0:00.11	/usr/lib/systemd/systemd-resolved
		660	systemd-ti	20	0	91024	7688	6784	S	0.0	0.2	0:00.07	/usr/lib/systemd/systemd-timesyncd
		713	systemd-ti	20	0	91024	7688	6784	S	0.0	0.2	0:00.00	/usr/lib/systemd/systemd-timesyncd
		754	root	20	0	53464	12158	10624	S	0.0	0.3	0:00.06	/usr/bin/vGAAuthService
		802	messagebus	20	0	9792	5376	4608	S	0.0	0.1	0:00.09	@dbus-daemon --system --system -address=systemd: --nofork --nopidfile --systemd-activation --syslog-o
		821	root	20	0	308M	9472	7536	S	0.0	0.2	0:00.00	/usr/bin/vmtoolsd
		825	polkitd	20	0	300M	7808	7648	S	0.0	0.2	0:00.04	/usr/lib/polkit-1/polkitd --no-debug
		841	root	20	0	308M	9472	7536	S	0.0	0.2	0:00.13	/usr/bin/vmtoolsd
		842	root	20	0	308M	9472	7536	S	0.0	0.2	0:00.00	/usr/bin/vmtoolsd
		849	root	20	0	18124	8832	7608	S	0.0	0.2	0:00.09	/usr/lib/systemd/systemd-logind
		853	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.09	/usr/libexec/udisks2/udisksd
		921	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.05	/usr/libexec/udisks2/udisksd
		923	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.00	/usr/libexec/udisks2/udisksd
		928	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.00	/usr/libexec/udisks2/udisksd
		956	root	20	0	107M	22912	13568	S	0.0	0.6	0:00.21	/usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
		963	syslog	20	0	217M	6016	4488	S	0.0	0.2	0:00.03	/usr/bin/rsyslog -n -NONE
		968	polkitd	20	0	300M	7808	7648	S	0.0	0.2	0:00.08	/usr/lib/polkit-1/polkitd --no-debug
		969	polkitd	20	0	300M	7808	7648	S	0.0	0.2	0:00.00	/usr/lib/polkit-1/polkitd --no-debug
		973	polkitd	20	0	300M	7808	7648	S	0.0	0.2	0:00.00	/usr/lib/polkit-1/polkitd --no-debug
		991	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.00	/usr/libexec/udisks2/udisksd
		992	root	20	0	382M	12928	10880	S	0.0	0.3	0:00.10	/usr/sbin/ModemManager
		1004	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.00	/usr/libexec/udisks2/udisksd
		1010	syslog	20	0	217M	6016	4488	S	0.0	0.2	0:00.00	/usr/bin/rsyslog -n -NONE
		1011	syslog	20	0	217M	6016	4488	S	0.0	0.2	0:00.00	/usr/bin/rsyslog -n -NONE
		1012	syslog	20	0	217M	6016	4488	S	0.0	0.2	0:00.00	/usr/bin/rsyslog -n -NONE
		1020	root	20	0	382M	12928	10880	S	0.0	0.3	0:00.00	/usr/sbin/ModemManager
		1023	root	20	0	382M	12928	10880	S	0.0	0.3	0:00.00	/usr/sbin/ModemManager
		1027	root	20	0	382M	12928	10880	S	0.0	0.3	0:00.00	/usr/sbin/ModemManager

### **6.5.5 Kết thúc tấn công**

Main I/O											
PID	USER	PRT	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
2518	ladadmin	20	0	8800	5120	3712	R	1.3	0.1	0:00.35	/sbin/init
1	root	20	0	22680	13832	9320		0.0	0.3	0:00.61	/sbin/init
450	root	19	-1	66608	15988	18064	S	0.0	0.5	0:00.00	/usr/lib/systemd/systemd-journald
502	root	RT	0	345M	12592	6074	S	0.0	0.0	0:00.50	/sbin/multipathd -d -s
511	root	20	0	30520	6590	6564	S	0.0	0.0	0:00.38	/usr/lib/systemd/systemd-udevd
517	root	20	0	345M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
518	root	RT	0	345M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
519	root	RT	0	345M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
520	root	RT	0	345M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
521	root	RT	0	345M	27392	8704	S	0.0	0.7	0:01.31	/sbin/multipathd -d -s
522	root	RT	0	345M	27392	8704	S	0.0	0.7	0:00.00	/sbin/multipathd -d -s
636	systemd-udevd	20	0	15008	9472	8320	S	0.0	0.2	0:00.00	/usr/lib/systemd/systemd-networkd
654	systemd-re	20	0	21580	12672	10496	S	0.0	0.3	0:00.14	/usr/lib/systemd/systemd-resolved
660	systemd-ti	20	0	91024	7688	6784	S	0.0	0.2	0:00.11	/usr/lib/systemd/systemd-timesyncd
713	systemd-ti	20	0	91024	7688	6784	S	0.0	0.2	0:00.00	/usr/lib/systemd/systemd-timesyncd
754	root	20	0	53464	12164	16624	S	0.0	0.3	0:00.00	/usr/bin/vAuthService
755	root	20	0	308M	9472	7936	S	0.0	0.2	0:25.33	/usr/bin/vmtoolsd
802	messagebus	20	0	9980	5376	4688	S	0.0	0.1	0:00.12	ibus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-o
821	root	20	0	308M	9472	7936	S	0.0	0.2	0:00.00	/usr/bin/vmtoolsd
825	polkitd	20	0	300M	7888	7040	S	0.0	0.2	0:00.85	/usr/lib/polkit-1/polkitd --no-debug
841	root	20	0	308M	9472	7936	S	0.0	0.2	0:00.41	/usr/bin/vmtoolsd
842	root	20	0	308M	9472	7936	S	0.0	0.2	0:00.00	/usr/bin/vmtoolsd
849	root	20	0	18124	8832	7608	S	0.0	0.2	0:00.10	/usr/lib/systemd/systemd-logind
853	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.10	/usr/libexec/udisks2/udisksd
921	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.15	/usr/libexec/udisks2/udisksd
923	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.00	/usr/libexec/udisks2/udisksd
928	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.00	/usr/libexec/udisks2/udisksd
956	root	20	0	107M	22912	13568	S	0.0	0.6	0:00.21	/usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
963	syslog	20	0	217M	6016	4480	S	0.0	0.2	0:00.03	/usr/sbin/syslogd -n -NONE
968	polkitd	20	0	300M	7888	7040	S	0.0	0.2	0:00.26	/usr/lib/polkit-1/polkitd --no-debug
969	polkitd	20	0	300M	7888	7040	S	0.0	0.2	0:00.00	/usr/lib/polkit-1/polkitd --no-debug
973	polkitd	20	0	300M	7888	7040	S	0.0	0.2	0:00.00	/usr/lib/polkit-1/polkitd --no-debug
991	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.00	/usr/libexec/ModemManager
992	root	20	0	382M	12528	10680	S	0.0	0.3	0:00.10	/usr/sbin/ModemManager
1000	root	20	0	457M	13568	11392	S	0.0	0.3	0:00.00	/usr/libexec/udisks2/udisksd
1010	syslog	20	0	217M	6016	4480	S	0.0	0.2	0:00.00	/usr/sbin/syslogd -n -NONE
1011	syslog	20	0	217M	6016	4480	S	0.0	0.2	0:00.00	/usr/sbin/syslogd -n -NONE
1012	syslog	20	0	217M	6016	4480	S	0.0	0.2	0:00.01	/usr/sbin/syslogd -n -NONE
1020	root	20	0	382M	12528	10680	S	0.0	0.3	0:00.00	/usr/libexec/ModemManager
1033	root	20	0	382M	12528	10680	S	0.0	0.3	0:00.00	/usr/sbin/ModemManager
1037	root	20	0	382M	12528	10680	S	0.0	0.3	0:00.00	/usr/sbin/ModemManager

### **6.5.6 Nhận xét**

- Ít SYN được chấp nhận hơn
  - Nhịp gói đều/giảm, không "bão" như mục 5
  - CPU/load ổn định hơn so với mục 5

→ Khi áp dụng SYN Cookies kết hợp với rate limiting bằng iptables, lưu lượng SYN Flood vẫn xuất hiện nhưng bị giới hạn về tốc độ. Hệ thống duy trì trạng thái ổn định.

CPU không tăng đáng kể và dịch vụ web vẫn truy cập được. Điều này cho thấy biện pháp phòng thủ hiệu quả đối với tấn công tầng giao vận (Layer 4)

### Kết luận:

Tiêu chí	SYN Flood (chưa phòng thủ)	SYN Flood (có phòng thủ)
Công cụ tấn công	hping3	hping3
Loại gói tin	TCP SYN	TCP SYN
Trạng thái kết nối	Nhiều half-open connection	Half-open giảm rõ
Hành vi server	Có giữ kết nối	Chủ động loại bỏ
Tài nguyên hệ thống	CPU/load tăng	Ôn định
Khả năng phục vụ	Suy giảm	Duy trì
Cơ chế phòng thủ	Không có	iptables, kernel
Hiệu quả	Thấp	Cao

→ Ở mục 5 và mục 6, báo cáo đã thực hiện tấn công TCP SYN Flood nhằm phân tích tác động của loại hình DoS ở tầng giao vận. Kết quả thực nghiệm cho thấy khi chưa có biện pháp phòng thủ, máy chủ dễ bị suy giảm khả năng phục vụ do số lượng lớn kết nối TCP ở trạng thái half-open. Sau khi áp dụng các biện pháp phòng chống, hệ thống duy trì trạng thái ổn định hơn

## 6.7 Tấn công HTTP FLOOD

- Mục tiêu:

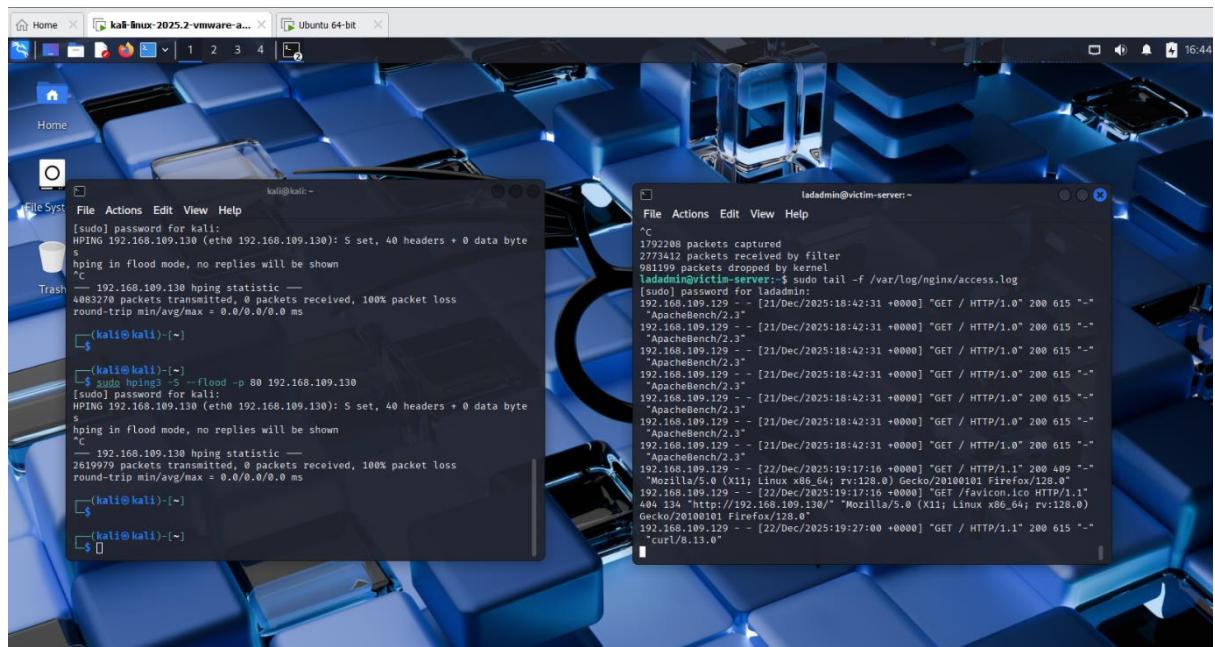
- + Tạo lưu lượng HTTP hợp lệ nhưng tần suất cao
  - + Quan sát tác động CPU / log / response time
- Không dùng iptables để chặn HTTP ở bước này

### 6.7.1. Chuẩn bị giám sát

- Terminal 1 - htop (Ubuntu local)

Main I/O											
PID	USER	PRT	NT	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
2516	ladadmin	20	0	8800	5120	3712	R	1.3	0.1	0:00:35	httpd
1	root	20	0	22080	15832	3920	S	0.0	0.3	0:03:61	/sbin/init
459	root	19	1	66832	15832	18000	S	0.0	0.5	0:01:00	/usr/lib/systemd/systemd-journald
502	root	RT	0	30529	27392	6784	S	0.0	0.7	0:00:50	/sbin/multipathd -d -s
511	root	20	0	30529	27392	8784	S	0.0	0.7	0:00:38	/usr/lib/systemd/systemd-udevd
517	root	20	0	3446	27392	8784	S	0.0	0.7	0:00:00	/sbin/multipathd -d -s
518	root	RT	0	3446	27392	8784	S	0.0	0.7	0:00:00	/sbin/multipathd -d -s
519	root	RT	0	3446	27392	8784	S	0.0	0.7	0:00:00	/sbin/multipathd -d -s
520	root	RT	0	3446	27392	8784	S	0.0	0.7	0:00:00	/sbin/multipathd -d -s
521	root	RT	0	3446	27392	8784	S	0.0	0.7	0:01:31	/sbin/multipathd -d -s
522	root	RT	0	3446	27392	8784	S	0.0	0.7	0:00:00	/sbin/multipathd -d -s
636	systemd-ne	20	0	19088	9472	8320	S	0.0	0.2	0:00:00	/usr/lib/systemd/systemd-networkd
654	systemd-re	20	0	21580	12672	10456	S	0.0	0.3	0:00:14	/usr/lib/systemd/systemd-resolved
660	systemd-ti	20	0	91024	7680	6784	S	0.0	0.2	0:00:11	/usr/lib/systemd/systemd-timesyncd
713	systemd-tl	20	0	91024	7680	6784	S	0.0	0.2	0:00:00	/usr/lib/systemd/systemd-timesyncd
754	root	20	0	53464	12160	10624	S	0.0	0.3	0:00:00	/usr/bin/VgAuthService
755	root	20	0	3008M	9472	7936	S	0.0	0.2	0:25:33	/usr/bin/vmtoolsd
802	messagebus	20	0	9300	5376	4688	S	0.0	0.1	0:00:12	@dbus-daemon --system --address=/system: --nofork --nopidfile --systemd-activation --syslog-o
821	root	20	0	3008M	9472	7936	S	0.0	0.2	0:00:00	/usr/bin/vmtoolsd
825	polkitd	20	0	3008M	7680	7040	S	0.0	0.2	0:00:05	/usr/lib/polkit-1/polkitd --no-debug
841	root	20	0	3008M	9472	7936	S	0.0	0.2	0:00:41	/usr/bin/vmtoolsd
842	root	20	0	3008M	9472	7936	S	0.0	0.2	0:00:00	/usr/bin/vmtoolsd
849	root	20	0	18124	8832	7888	S	0.0	0.2	0:00:10	/usr/lib/systemd/systemd-logind
853	root	20	0	457M	13568	11392	S	0.0	0.3	0:00:10	/usr/libexec/udisks2/udisksd
921	root	20	0	457M	13568	11392	S	0.0	0.3	0:00:15	/usr/libexec/udisks2/udisksd
923	root	20	0	457M	13568	11392	S	0.0	0.3	0:00:00	/usr/libexec/udisks2/udisksd
928	root	20	0	457M	13568	11392	S	0.0	0.3	0:00:00	/usr/libexec/udisks2/udisksd
956	root	20	0	167M	22912	13568	S	0.0	0.5	0:01:21	/usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
963	syslog	20	0	217M	6016	4480	S	0.0	0.2	0:00:03	/usr/sbin/syslogd -n -NONE
966	polkitd	20	0	3008M	7680	7040	S	0.0	0.2	0:00:26	/usr/lib/polkit-1/polkitd --no-debug
969	polkitd	20	0	3008M	7680	7040	S	0.0	0.2	0:00:00	/usr/lib/polkit-1/polkitd --no-debug
973	polkitd	20	0	3008M	7680	7040	S	0.0	0.2	0:00:00	/usr/lib/polkit-1/polkitd --no-debug
991	root	20	0	457M	13568	11392	S	0.0	0.3	0:00:00	/usr/libexec/udisks2/udisksd
992	root	20	0	382M	12928	10880	S	0.0	0.3	0:00:10	/usr/sbin/ModemManager
1004	root	20	0	457M	13568	11392	S	0.0	0.3	0:00:00	/usr/libexec/udisks2/udisksd
1010	syslog	20	0	217M	6016	4480	S	0.0	0.2	0:00:00	/usr/sbin/syslogd -n -NONE
1011	syslog	20	0	217M	6016	4480	S	0.0	0.2	0:00:00	/usr/sbin/syslogd -n -NONE
1012	syslog	20	0	217M	6016	4480	S	0.0	0.2	0:00:01	/usr/sbin/syslogd -n -NONE
1020	root	20	0	382M	12928	10880	S	0.0	0.3	0:00:00	/usr/sbin/ModemManager
1023	root	20	0	382M	12928	10880	S	0.0	0.3	0:00:00	/usr/sbin/ModemManager
1027	root	20	0	382M	12928	10880	S	0.0	0.3	0:00:00	/usr/sbin/ModemManager

- Terminal 2 - theo dõi access log (Ubuntu SSH)



### 6.7.2 Chuẩn bị tấn công (Kali)

- Mở terminal mới trên Kali Linux, dùng ab (ApacheBench) để tấn công

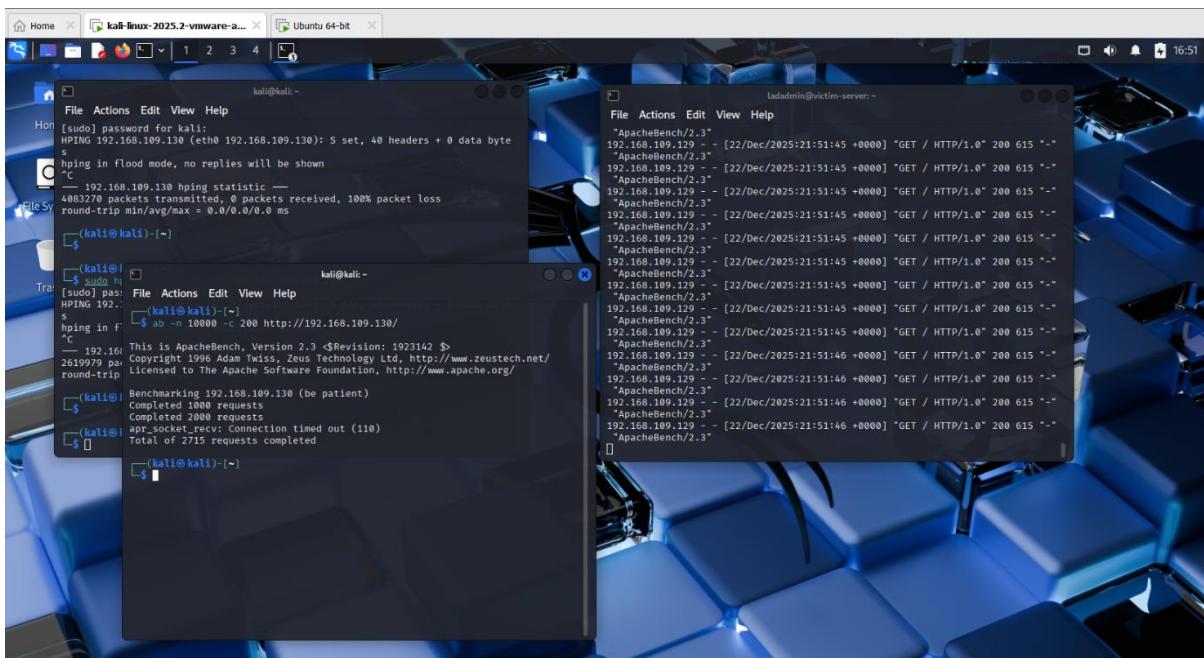
```
ab -n 10000 -c 200 http://<IP UBUNTU>/
```

-n 10000: tổng số request

-c 200: 200 request song song

URL / là endpoint web

### 6.7.3 Chạy HTTP Flood



```

0[|]
1[|]
Mem[|||||] 0K/0K
Sup[|]

Main I/O PID USER PRI NI VIRT RES SHR S CPU%+MEM% TIME+ Command
1178 www-data 20 0 12800 4916 3584 S 1.3 0.1 0:00:11 nginx: worker process
755 root 20 0 380M 9472 7936 S 0.7 0.2 0:26.17 /usr/bin/vmtoolsd
2518 ladmin 20 0 8800 5120 3712 R 0.7 0.1 0:02.48 httpd
2523 root 20 0 5716 1792 1792 S 0.7 0.0 0:00:03 tail -f /var/log/nginx/access.log
1 root 20 0 22000 13032 9320 S 0.0 0.3 0:03.61 /sbin/init
450 root 19 -1 66832 19688 16664 S 0.0 0.5 0:00:35 /usr/lib/systemd/systemd-journald
502 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:51 /sbin/multipathd -d -s
511 root 20 0 30520 8564 4664 S 0.0 0.2 0:00:38 /usr/lib/systemd/systemd-udevd
517 root 20 0 346M 27392 8704 S 0.0 0.7 0:00:06 /sbin/multipathd -d -s
518 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:06 /sbin/multipathd -d -s
519 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:06 /sbin/multipathd -d -s
520 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:06 /sbin/multipathd -d -s
521 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:06 /sbin/multipathd -d -s
522 root RT 0 346M 27392 8704 S 0.0 0.7 0:00:06 /sbin/multipathd -d -s
636 systemd-ne 20 0 19000 9472 6304 S 0.0 0.2 0:00:06 /usr/lib/systemd/systemd-networkd
654 systemd-re 20 0 21580 12672 1192 S 0.0 0.1 0:00:11 /usr/lib/systemd/systemd-resolved
660 systemd-t 20 0 91024 7680 5784 S 0.0 0.2 0:00:11 /usr/lib/systemd/systemd-timesyncd
713 systemd-t 20 0 91024 7680 5784 S 0.0 0.2 0:00:00 /usr/lib/systemd/systemd-timesyncd
754 root 20 0 50164 12160 10624 S 0.0 0.3 0:00:06 /usr/bin/VGAuthService
802 messagebus 20 0 9200 5376 4608 S 0.0 0.1 0:00:12 @dbus-daemon system --address=system: --nofork --nopidfile --systemd-activation --syslog-o
821 root 20 0 308M 9472 7936 S 0.0 0.2 0:00:00 /usr/lib/systemd/elogind
825 polkitd 20 0 300M 7808 7040 S 0.0 0.2 0:00:05 /usr/lib/polkit-1/polkitd --no-debug
841 root 20 0 308M 9472 7936 S 0.0 0.2 0:00:42 /usr/bin/mmcold
842 root 20 0 308M 9472 7936 S 0.0 0.2 0:00:00 /usr/bin/vmcold
849 root 20 0 18124 8832 7808 S 0.0 0.2 0:00:10 /usr/lib/systemd/systemd-logind
853 root 20 0 457M 13568 11392 S 0.0 0.3 0:00:10 /usr/libexec/udisks2/udisksd
921 root 20 0 457M 13568 11392 S 0.0 0.3 0:00:15 /usr/libexec/udisks2/udisksd
923 root 20 0 457M 13568 11392 S 0.0 0.3 0:00:00 /usr/libexec/udisks2/udisksd
928 root 20 0 457M 13568 11392 S 0.0 0.3 0:00:00 /usr/libexec/udisks2/udisksd
956 root 20 0 107M 22912 13568 S 0.0 0.6 0:00:21 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
963 syslog 20 0 217M 5016 4480 S 0.0 0.2 0:00:03 /usr/sbin/rsyslogd -n -NONE
968 polkitd 20 0 300M 7808 7040 S 0.0 0.2 0:00:27 /usr/lib/polkit-1/polkitd --no-debug
969 polkitd 20 0 300M 7808 7040 S 0.0 0.2 0:00:00 /usr/lib/polkit-1/polkitd --no-debug
973 polkitd 20 0 300M 7808 7040 S 0.0 0.2 0:00:00 /usr/lib/polkit-1/polkitd --no-debug
991 root 20 0 457M 13568 11392 S 0.0 0.3 0:00:00 /usr/libexec/udisks2/udisksd
992 root 20 0 382M 12928 10880 S 0.0 0.3 0:00:10 /usr/sbin/ModemManager
1004 root 20 0 457M 13568 11392 S 0.0 0.3 0:00:00 /usr/libexec/udisks2/udisksd
1010 syslog 20 0 217M 5016 4480 S 0.0 0.2 0:00:00 /usr/sbin/rsyslogd -n -NONE
1011 syslog 20 0 217M 5016 4480 S 0.0 0.2 0:00:00 /usr/sbin/rsyslogd -n -NONE
1012 syslog 20 0 217M 5016 4480 S 0.0 0.2 0:00:01 /usr/sbin/rsyslogd -n -NONE
1020 root 20 0 382M 12928 10880 S 0.0 0.3 0:00:00 /usr/sbin/ModemManager
F:Help F8Setup F8Search F5Tree F6SortByFNice F8Nice F9Kill F10Quit

```

### 6.7.4 Kết thúc tấn công

Đợi ab chạy xong (hoặc Ctrl + C nếu cần)

- CPU giảm dần

- Hệ thống hồi phục

### 6.7.5 Nhận xét

Khi ab đang chạy:

- CPU tăng rõ
- Load average tăng
- Nginx worker hoạt động nhiều
- Log dày đặc
- Nhiều dòng HTTP 200 trong thời gian ngắn

→ HTTP Flood sử dụng các request hợp lệ ở tầng ứng dụng, gây tăng tần suất truy cập và làm access log tăng nhanh. Mặc dù mức sử dụng CPU không tăng đột biến, server xuất hiện timeout khi xử lý số lượng lớn kết nối đồng thời, cho thấy dịch vụ bị suy giảm hiệu năng

## 6.8 Phòng thủ HTTP FLOOD (Layer 7)

- Mục tiêu:
  - + Hạn chế số request/giây theo IP
  - + Giảm timeout và ổn định response time
  - + So sánh trước và sau khi phòng thủ

### 6.8.1 Mở file cấu hình Nginx (Ubuntu)

```

GNU nano 7.2
/etc/nginx/nginx.conf

user www-data;
worker_processes auto;
pid /run/nginx.pid;
error_log /var/log/nginx/error.log;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    ##
    # Logging Settings
    ##

    access_log /var/log/nginx/access.log;

    ##
    # Gzip Settings
    ##

    gzip on;
}

[ Read 83 lines ]

```

GNU nano 7.2      /etc/nginx/nginx.conf

user www-data;  
worker\_processes auto;  
pid /run/nginx.pid;  
error\_log /var/log/nginx/error.log;  
include /etc/nginx/modules-enabled/\*.conf;

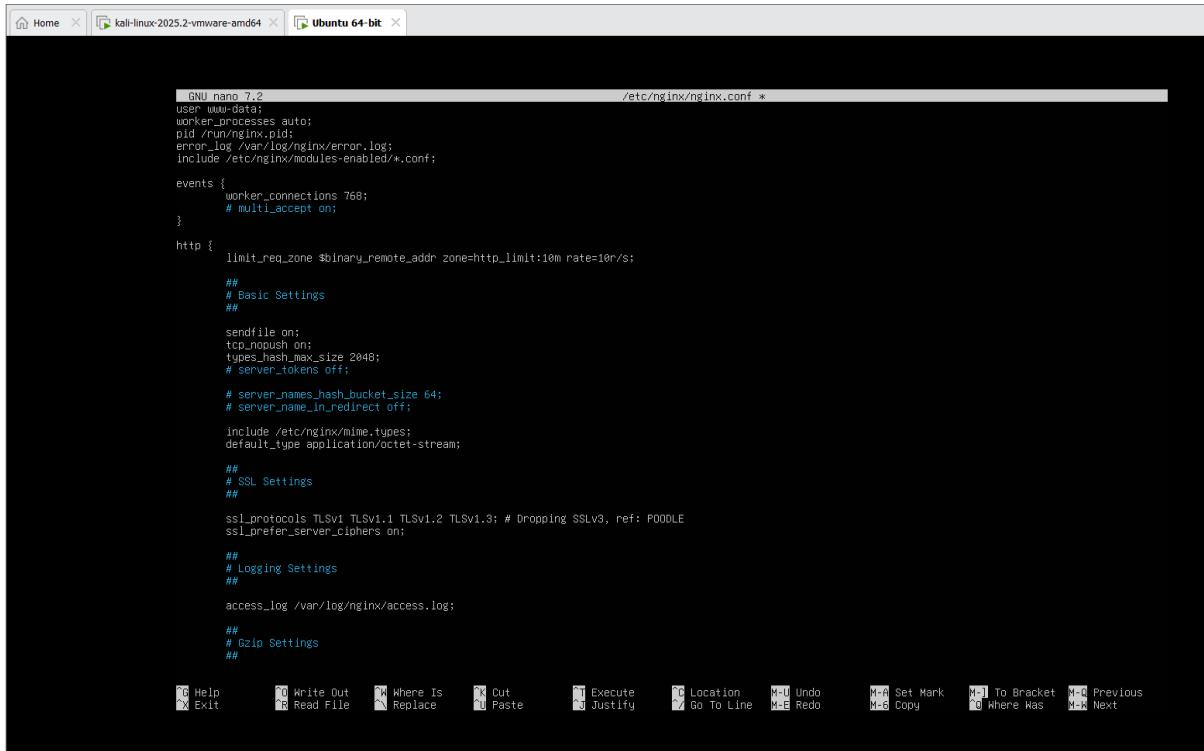
events {  
 worker\_connections 768;  
 # multi\_accept on;  
}

http {  
 ##  
 # Basic Settings  
 ##  
  
 sendfile on;  
 tcp\_nopush on;  
 types\_hash\_max\_size 2048;  
 # server\_tokens off;  
  
 # server\_names\_hash\_bucket\_size 64;  
 # server\_name\_in\_redirect off;  
  
 include /etc/nginx/mime.types;  
 default\_type application/octet-stream;  
  
 ##  
 # SSL Settings  
 ##  
  
 ssl\_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE  
 ssl\_prefer\_server\_ciphers on;  
  
 ##  
 # Logging Settings  
 ##  
  
 access\_log /var/log/nginx/access.log;  
  
 ##  
 # Gzip Settings  
 ##  
  
 gzip on;  
}

[ Read 83 lines ]

G Help      ^O Write Out      ^W Where Is      ^K Cut      T Execute      ^C Location      M-U Undo  
X Exit      ^R Read File      ^L Replace      ^U Paste      J Justify      ^Y Go To Line      M-E Redo  
M-A Set Mark      M-J To Bracket      M-Q Where Was      M-Q Previous  
M-B Copy      M-Q Where Was      M-W Next

## 6.8.2 Thêm zone giới hạn request (HTTP context)



```
GNU nano 7.2                                         /etc/nginx/nginx.conf *
user www-data;
worker_processes auto;
pid /run/nginx.pid;
error_log /var/log/nginx/error.log;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
    limit_req_zone $binary_remote_addr zone=http_limit:10m rate=10r/s;

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    ##
    # Logging Settings
    ##

    access_log /var/log/nginx/access.log;

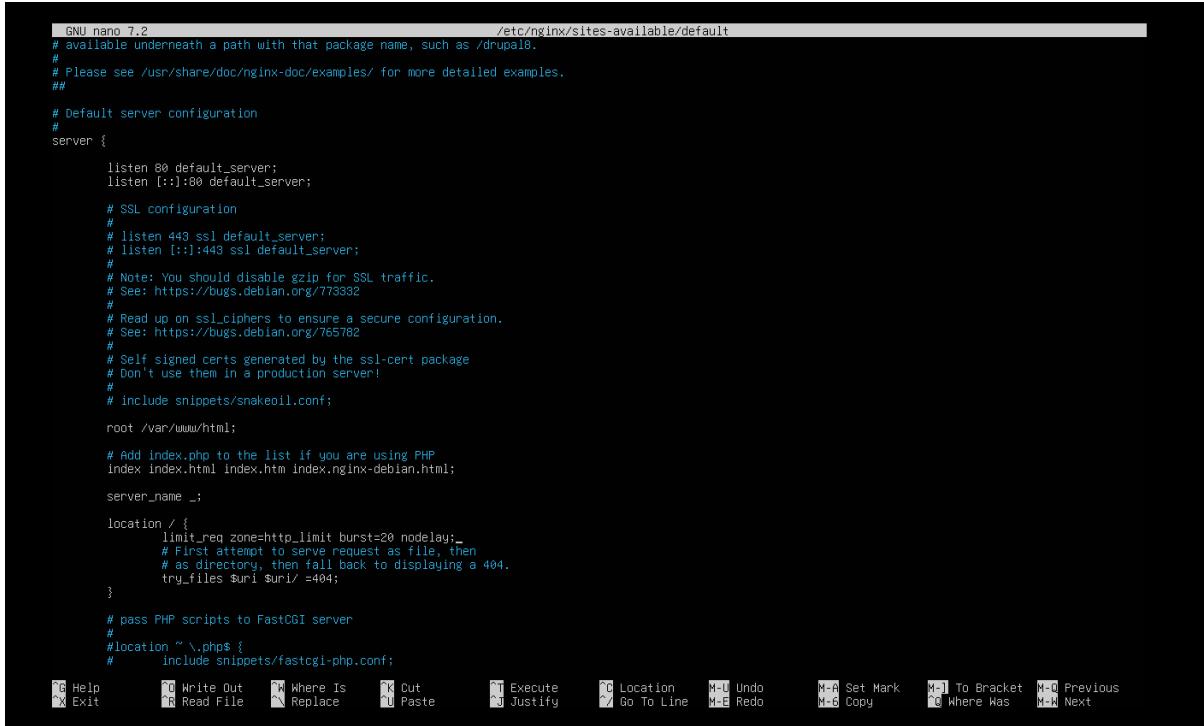
    ##
    # Gzip Settings
    ##

    gzip on;
    gzip_disable "msie6";
    gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_buffers 16 8k;
    gzip_types text/plain text/css application/json application/javascript
               application/javascript;
    gzip_min_length 1024;
}
```

10r/s: tối đa 10 request/giây/IP

10m: đủ cho ~160k IP (lab dù)

## 6.8.3 Áp dụng limit cho server (server/location)



```
GNU nano 7.2                                         /etc/nginx/sites-available/default
# available underneath a path with that package name, such as /drupal18.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {

    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        limit_req zone=http_limit burst=20 nodelay;
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }

    # pass PHP scripts to FastCGI server
    #
    #location ~ \.php$ {
    #    include snippets/fastcgi-php.conf;
    #}
}
```

- burst=20: cho phép bùng nổ ngắn

- nodelay: không xếp hàng, vượt ngưỡng là trả lỗi sớm

## 6.8.4 Kiểm tra & reload Nginx (Không restart)

```
ladadmin@victim-server:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ladadmin@victim-server:~$ sudo systemctl reload nginx
ladadmin@victim-server:~$
```

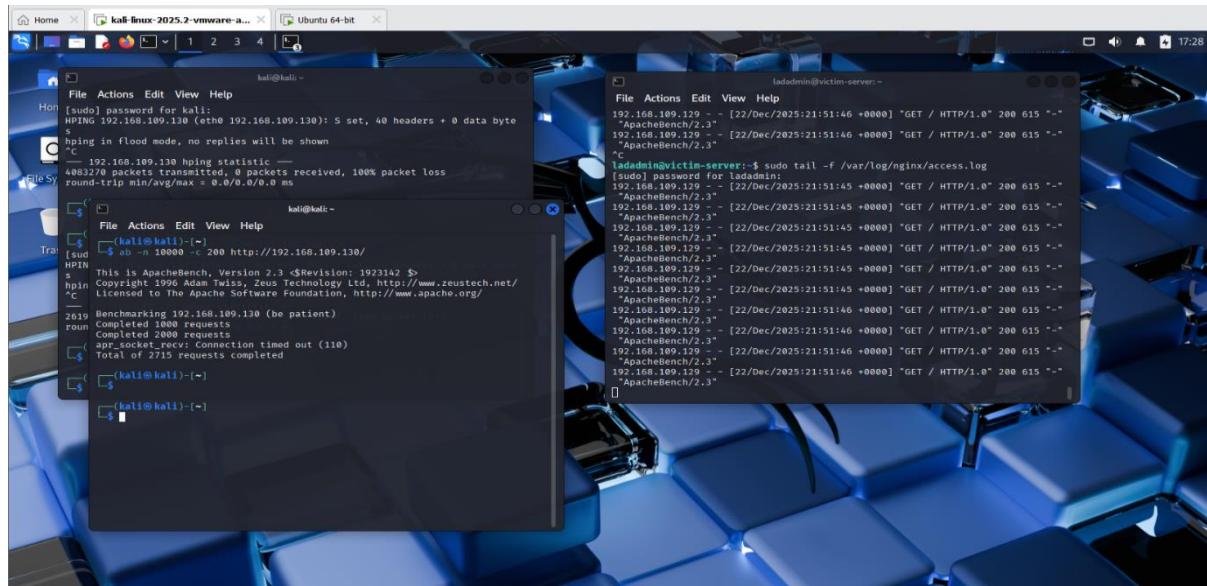
## 6.8.5 Chuẩn bị giám sát

- Ubuntu local:

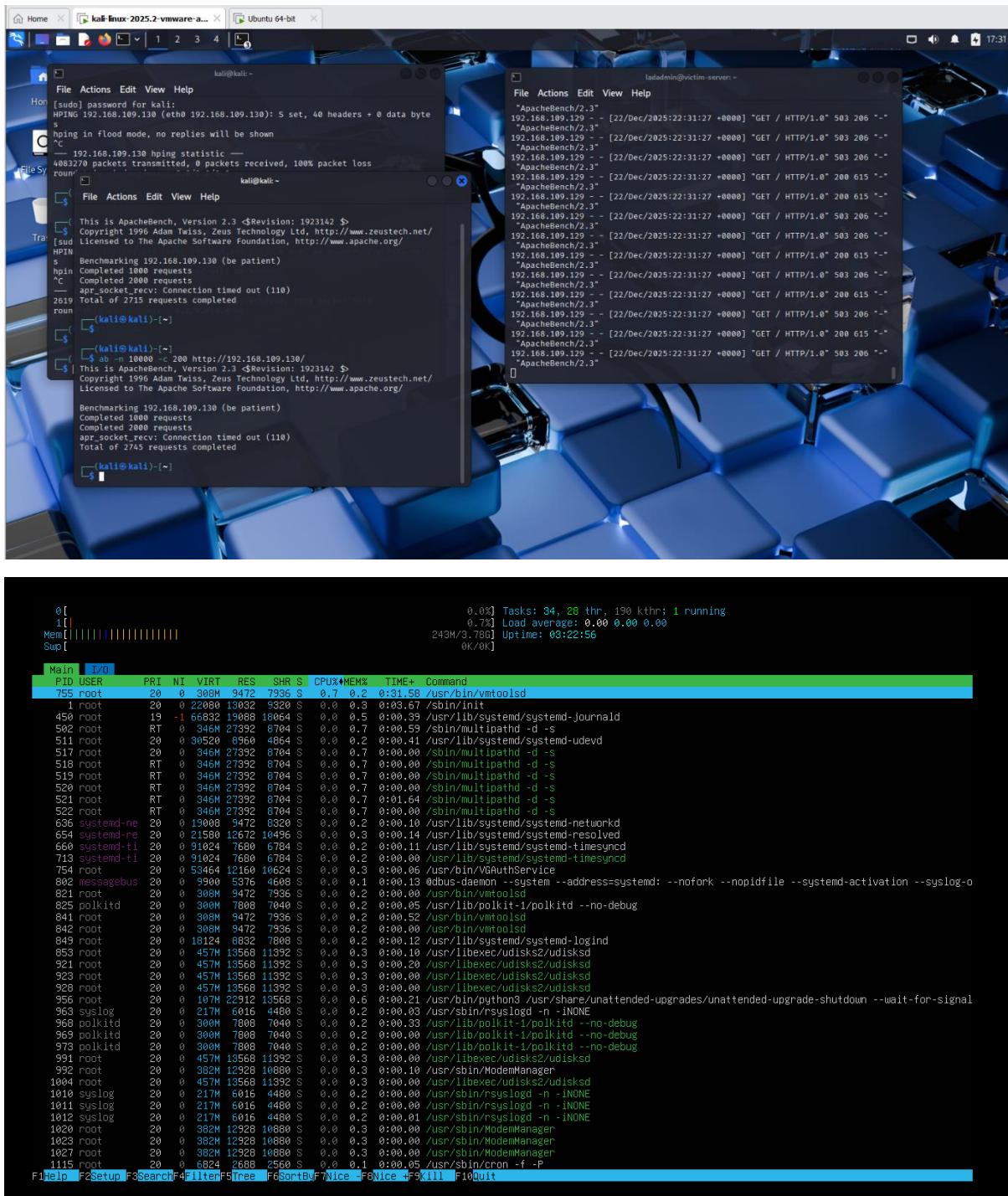
The terminal window displays the following information:

- System Status:** Tasks: 30, 20 thr, 190 kthr; 1 running, 0.7% Load average: 0.00 0.00 0.00, 238M/3.76G Uptime: 03:16:44, 0K/0K.
- Memory Usage:** Mem: 1.0G used, 1.0G free, 1.0G available, 0K/0K.
- Process List:** A detailed list of processes including:
  - 2619 ladadmin (root) 20 0 8800 5120 3712 R 0.7 0.1 0:00.04 htop
  - 1 root (root) 20 0 22689 13092 9320 S 0.0 0.3 0:00.38 /sbin/init
  - 450 root (root) 19 1 66289 15088 10088 S 0.0 0.3 0:00.38 /usr/lib/systemd/systemd-journald
  - 502 root (RT) 20 0 346M 2169 8704 S 0.0 0.7 0:00.57 /sbin/multipathd -d -s
  - 511 root (RT) 20 0 30520 3556 4864 S 0.0 0.2 0:00.41 /usr/lib/systemd/systemd-udevd
  - 517 root (RT) 20 0 345M 27392 8704 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
  - 518 root (RT) 20 0 345M 27392 8704 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
  - 519 root (RT) 20 0 345M 27392 8704 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
  - 520 root (RT) 20 0 345M 27392 8704 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
  - 521 root (RT) 20 0 345M 27392 8704 S 0.0 0.7 0:01.53 /sbin/multipathd -d -s
  - 522 root (RT) 20 0 345M 27392 8704 S 0.0 0.7 0:00.00 /sbin/multipathd -d -s
  - 636 systemd-ne (root) 20 0 19008 5472 8292 S 0.0 0.2 0:00.18 /usr/lib/systemd/systemd-networkd
  - 654 systemd-re (root) 20 0 21580 12672 10496 S 0.0 0.3 0:00.14 /usr/lib/systemd/systemd-resolved
  - 660 systemd-ti (root) 20 0 91024 7680 5784 S 0.0 0.2 0:00.11 /usr/lib/systemd/systemd-timesyncd
  - 713 systemd-ti (root) 20 0 91024 7680 5784 S 0.0 0.2 0:00.00 /usr/lib/systemd/systemd-timesyncd
  - 754 root (root) 20 0 53464 12169 10624 S 0.0 0.3 0:00.06 /usr/bin/vGAAuthService
  - 755 root (root) 20 0 308M 9472 7936 S 0.0 0.2 0:00.74 /usr/bin/vmtoolsd
  - 802 messagebus (root) 20 0 9500 5376 4608 S 0.0 0.1 0:00.13 @dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-o
  - 821 root (root) 20 0 308M 9472 7936 S 0.0 0.2 0:00.00 /usr/bin/vmtoolsd
  - 825 polkitd (root) 20 0 300M 7808 7048 S 0.0 0.2 0:00.05 /usr/lib/polkit-1/polkitd --no-debug
  - 841 root (root) 20 0 300M 9472 7936 S 0.0 0.2 0:00.58 /usr/bin/vmtoolsd
  - 842 root (root) 20 0 300M 9472 7936 S 0.0 0.2 0:00.00 /usr/bin/vmtoolsd
  - 849 root (root) 20 0 18124 8832 7808 S 0.0 0.2 0:00.12 /usr/lib/systemd/systemd-logind
  - 853 root (root) 20 0 457M 13568 11392 S 0.0 0.3 0:00.18 /usr/libexec/udisks2/udisksd
  - 921 root (root) 20 0 457M 13568 11392 S 0.0 0.3 0:00.28 /usr/libexec/udisks2/udisksd
  - 923 root (root) 20 0 457M 13568 11392 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
  - 928 root (root) 20 0 457M 13568 11392 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
  - 956 root (root) 20 0 107M 22912 13568 S 0.0 0.6 0:00.21 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
  - 963 syslog (root) 20 0 217M 6016 4488 S 0.0 0.2 0:00.03 /usr/sbin/rsyslogd -n -INONE
  - 968 polkitd (root) 20 0 300M 7808 7048 S 0.0 0.2 0:00.32 /usr/lib/polkit-1/polkitd --no-debug
  - 969 polkitd (root) 20 0 300M 7808 7048 S 0.0 0.2 0:00.00 /usr/lib/polkit-1/polkitd --no-debug
  - 973 polkitd (root) 20 0 300M 7808 7048 S 0.0 0.2 0:00.00 /usr/lib/polkit-1/polkitd --no-debug
  - 991 root (root) 20 0 457M 13568 11392 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
  - 992 root (root) 20 0 382M 12928 10880 S 0.0 0.3 0:00.18 /usr/sbin/ModemManager
  - 1004 root (root) 20 0 457M 13568 11392 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
  - 1010 syslog (root) 20 0 217M 6016 4488 S 0.0 0.2 0:00.00 /usr/sbin/rsyslogd -n -INONE
  - 1011 syslog (root) 20 0 217M 6016 4488 S 0.0 0.2 0:00.00 /usr/sbin/rsyslogd -n -INONE
  - 1012 syslog (root) 20 0 217M 6016 4488 S 0.0 0.2 0:00.01 /usr/sbin/rsyslogd -n -INONE
  - 1020 root (root) 20 0 382M 12928 10880 S 0.0 0.3 0:00.00 /usr/sbin/ModemManager
  - 1023 root (root) 20 0 382M 12928 10880 S 0.0 0.3 0:00.00 /usr/sbin/ModemManager
  - 1027 root (root) 20 0 382M 12928 10880 S 0.0 0.3 0:00.00 /usr/sbin/ModemManager

- Ubuntu SSH:



### **6.8.6 Chạy lại HTTP Flood (Kali)**



### 6.8.7 Dùng

Đợi ab xong hoặc Ctrl + C

### 6.8.8 Nhận xét

- Ở access.log : Thấy ít dòng hơn, có thể xuất hiện 429 Too Many Requests

- Ở htop: CPU/load ổn định hơn trước khi phòng thủ, Nginx worker không bị "dồn", không còn dấu hiệu quá tải và ổn định rõ rệt so với khi không phòng thủ → Phòng thủ thành công về mặt tài nguyên

- Ở kali - ab: Số request hoàn thành ít hơn, ít timeout hơn hoặc kết thúc sớm do limit  
 → Sau khi áp dụng rate limiting tại Nginx, HTTP Flood bị hạn chế hiệu quả. Máy chủ duy trì ổn định hơn, thời gian phản hồi cải thiện và xuất hiện mã 429 đối với các request vượt ngưỡng. Biện pháp này giúp bảo vệ tầng ứng dụng trước các tấn công DoS/DDoS quy mô nhỏ - trung bình → hiệu quả ở Layer 7 nhưng cần tinh chỉnh ngưỡng để tránh ảnh hưởng người dùng hợp lệ

### Kết luận kỹ thuật:

Tiêu chí	HTTP Flood (chưa phòng thủ)	HTTP Flood (có phòng thủ)
Số request xử lý	Cao, dồn dập	Bị giới hạn
Timeout	Có (do quá tải)	Có (do bị chặn)
HTTP status	Chủ yếu 200	Xuất hiện 503
CPU / Load	Tăng nhẹ, worker bận	Ôn định
Khả năng phục vụ	Suy giảm	Duy trì ổn định

→ Rate limiting tại Nginx hoạt động đúng mục tiêu

→ Sau khi triển khai cơ chế rate limiting tại Nginx, HTTP Flood bị hạn chế hiệu quả. Các request vượt ngưỡng bị từ chối với mã phản hồi 503, trong khi các request hợp lệ vẫn được xử lý. So với trạng thái trước phòng thủ, hệ thống duy trì mức sử dụng CPU và load ổn định hơn, cho thấy biện pháp này hiệu quả trong việc bảo vệ dịch vụ ở tầng ứng dụng (Layer 7)

## KẾT LUẬN

Tiêu luận đã tổng hợp và phân tích toàn diện về tấn công từ chối dịch vụ (DoS) và tấn công từ chối dịch vụ phân tán (DDoS) — những mối đe dọa chủ yếu đối với an ninh hệ thống mạng trong thời đại số hiện nay. Qua nghiên cứu, ta đã làm rõ bản chất và nguyên lý hoạt động chung của DoS/DDoS là gây quá tải tài nguyên mạng hoặc dịch vụ bằng lưu lượng giả mạo lớn nhằm *ngăn chặn người dùng hợp pháp truy cập tài nguyên*. DoS thường phát sinh từ *một nguồn duy nhất*, trong khi DDoS sử dụng *nhiều nguồn phân tán* (botnet) khiến việc phát hiện và phòng chống trở nên phức tạp hơn nhiều. Các kỹ thuật tấn công phổ biến như *SYN Flood*, *Ping of Death* hay *DNS Amplification* đều khai thác điểm yếu trong các giao thức và quy trình xử lý của hệ thống để đạt được mục tiêu này. Hậu quả của các cuộc tấn công DoS/DDoS không chỉ là gián đoạn dịch vụ, tiêu tốn tài nguyên hệ thống, làm sụt giảm hiệu suất hoạt động, mà còn gây thiệt hại về tài chính, mất uy tín doanh nghiệp và rủi ro bảo mật dữ liệu. Vì vậy, hiểu rõ các đặc tính tấn công, xây dựng mô hình phòng chống ở nhiều tầng (mạng, hệ điều hành, ứng dụng) và sử dụng giải pháp chuyên dụng là điều quan trọng để bảo đảm tính sẵn sàng, ổn định của hệ thống mạng trong tương lai. (DoS/DDoS đều là loại tấn công gây quá tải tài nguyên và

ngăn chặn truy cập hợp lệ; nghiên cứu cũng chỉ ra đặc tính và biện pháp phòng ngừa phù hợp)

## TÀI LIỆU THAM KHẢO

- [2.1] Thế Giới Di Động (2023), *DoS, DDoS là gì? Nhận biết, ngăn chặn tấn công từ chối dịch vụ*, Thế Giới Di Động Game/App. [Truy cập ngày 20/12/2025]
- [2.2] Sunteco (2025), *DoS là gì? Cách nhận biết tấn công DoS và bảo vệ hệ thống*, Sunteco Blog. [Truy cập ngày 20/12/2025]
- [2.3] VNETWORK (2025), *DDoS là gì? Cách ngăn chặn các loại tấn công DDoS Server*, VNETWORK Security News. [Truy cập ngày 20/12/2025]
- [3.1] Cloudflare (2024), *Ping of death DDoS attack*, Cloudflare Learning Center. [Truy cập ngày 20/12/2025]
- [3.2] Fortinet (2024), *What Is A Ping Of Death Attack?*, Fortinet Cyber Glossary. [Truy cập ngày 20/12/2025]
- [3.3] USENIX (2018), *Fragmentation Considered Vulnerable*, USENIX Security Symposium. [Truy cập ngày 20/12/2025]
- [3.4] Radware (2024), *What is Ping of Death (PoD) Attack?*, Radware Security Knowledge Base. [Truy cập ngày 20/12/2025]
- [3.5] Imperva (2024), *What is Ping of Death (PoD) | Prevention & Mitigation Methods*, Imperva Learning Center. [Truy cập ngày 20/12/2025]
- [3.6] NIST (1999), *CVE-1999-0128 – Ping of Death*, National Vulnerability Database. [Truy cập ngày 20/12/2025]
- [3.7] Blumira (2020), *Ping of Death v2 Windows IPv6 Vulnerability CVE-2020-16898/9*, Blumira Security Blog. [Truy cập ngày 20/12/2025]
- [3.8] Juniper Networks (2023), *Configuring Ping-Of-Death Attack Screen*, Juniper TechLibrary. [Truy cập ngày 20/12/2025]
- [3.9] Atomic Wallet (2023), *Teardrop Attack Explained – Understanding Packet Fragmentation*, Atomic Wallet Academy. [Truy cập ngày 20/12/2025]
- [3.10] Wallarm (2024), *What is an IP Fragmentation Attack? Detailed Overview*, Wallarm Learning. [Truy cập ngày 20/12/2025]
- [3.11] Startup Defense (2024), *Ping of Death Attacks: History Impact and Prevention*, Startup Defense Blog. [Truy cập ngày 20/12/2025]
- [3.12] Cloudflare (2024), *SYN flood DDoS attack*, Cloudflare Learning Center. [Truy cập ngày 20/12/2025]
- [3.13] Wikipedia (2024), *SYN flood*, Wikimedia Foundation. [Truy cập ngày 20/12/2025]

- [3.14] Radware (2024), *What Are TCP SYN Flood DDoS Attacks & 6 Ways to Stop Them*, Radware Security. [Truy cập ngày 20/12/2025]
- [3.15] Imperva (2024), *What is a TCP SYN Flood | Mitigation Techniques*, Imperva Learning Center. [Truy cập ngày 20/12/2025]
- [3.16] JumpCloud (2023), *What Is a SYN Cookie? TCP Security Mechanism Explained*, JumpCloud Blog. [Truy cập ngày 20/12/2025]
- [3.17] Wikipedia (2024), *SYN cookies*, Wikimedia Foundation. [Truy cập ngày 20/12/2025]
- [3.18] D. J. Bernstein (1996), *SYN cookies*, cr.yp.to. [Truy cập ngày 20/12/2025]
- [3.19] SonicWall (2023), *SonicOS Firewall – SYN Proxy*, SonicWall Technical Documentation. [Truy cập ngày 20/12/2025]
- [3.20] KLAS Telecom (2024), *Mitigating TCP Flood Attacks Using the SYNPROXY Feature*, KLAS Documentation. [Truy cập ngày 20/12/2025]
- [3.21] Indusface (2024), *DNS Amplification Attack*, Indusface Learning. [Truy cập ngày 20/12/2025]
- [3.22] CISA (2013), *Alert (TA13-088A): DNS Amplification Attacks*, Cybersecurity and Infrastructure Security Agency. [Truy cập ngày 20/12/2025]
- [3.23] NETSCOUT (2025), *DDoS Threat Intelligence Report 2025*, NETSCOUT Systems. [Truy cập ngày 20/12/2025]
- [3.24] DeepStrike (2025), *DDoS Attack Statistics: 20.5M Attacks Blocked in Q1 2025*, DeepStrike Security Report. [Truy cập ngày 20/12/2025]
- [3.25] Imperva (2025), *Early 2025 DDoS Attacks Signal a Dangerous Trend*, Imperva Blog. [Truy cập ngày 20/12/2025]
- [3.26] DDoS-Guard (2025), *L7 Is Pulling Ahead – DDoS Trends 2025*, DDoS-Guard Blog. [Truy cập ngày 20/12/2025]
- [3.27] FastNetMon (2025), *DDoS Attacks 2025 Trends and Mitigation*, FastNetMon Blog. [Truy cập ngày 20/12/2025]
- [4.1] ENISA (2023), *ENISA Threat Landscape 2023*, European Union Agency for Cybersecurity. [Truy cập ngày 20/12/2025]
- [4.2] Cisco (2024), *What Is a Botnet?*, Cisco Security Concepts. [Truy cập ngày 20/12/2025]
- [4.3] NIST (2012), *Computer Security Incident Handling Guide (SP 800-61 Rev.2)*, National Institute of Standards and Technology. [Truy cập ngày 20/12/2025]
- [4.4] Cloudflare (2024), *What is a botnet?*, Cloudflare Learning Center. [Truy cập ngày 20/12/2025]

- [4.5] Akamai (2023), *State of the Internet / Security Report*, Akamai Technologies. [Truy cập ngày 20/12/2025]
- [5.2] IETF (2000), *RFC 2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*, Internet Engineering Task Force. [Truy cập ngày 20/12/2025]
- [5.3] Cisco (2024), *Configure Commonly Used IP ACLs*, Cisco Support Documentation. [Truy cập ngày 20/12/2025]
- [5.4] IETF (2009), *RFC 5635: Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)*, Internet Engineering Task Force. [Truy cập ngày 20/12/2025]
- [5.5] IETF (2007), *RFC 4987: TCP SYN Flooding Attacks and Common Mitigations*, Internet Engineering Task Force. [Truy cập ngày 20/12/2025]
- [5.6] Red Hat (2024), *Kernel Tuning Guide: Countering a SYN Flood*, Red Hat Customer Portal. [Truy cập ngày 20/12/2025]
- [5.7] OWASP (2024), *OWASP ModSecurity Core Rule Set (CRS)*, OWASP Foundation. [Truy cập ngày 20/12/2025]
- [5.8] Google (2024), *What is reCAPTCHA?*, Google Developers. [Truy cập ngày 20/12/2025]
- [5.9] NGINX (2023), *Mitigating DDoS Attacks with NGINX and NGINX Plus*, NGINX Blog. [Truy cập ngày 20/12/2025]
- [5.10] Cloudflare (2024), *What is a CDN?*, Cloudflare Learning Center. [Truy cập ngày 20/12/2025]
- [5.11] NIST (2019), *SP 800-189: Resilient Interdomain Traffic Exchange: BGP Security and DDoS Mitigation*, National Institute of Standards and Technology. [Truy cập ngày 20/12/2025]