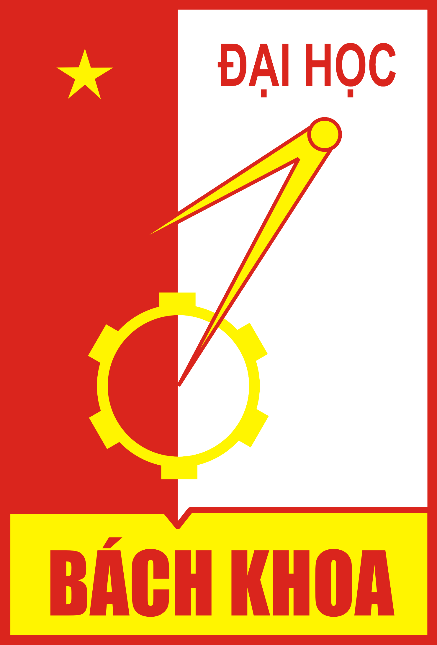
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Viện Công nghệ Thông tin và Truyền thông**



**BÁO CÁO BÀI TẬP LỚN**

Môn học**: An ninh mạng**

Đề tài: **10 - Tìm hiểu tấn công DoS/DDos trong mạng và**

**cách thức phòng chống.**

Giảng viên hướng dẫn: **ThS.** **Bùi Trọng Tùng**

Lớp: **LTU15**

Nhóm: **6**

Thành viên: 1. Nguyễn Đức Thiên MSSV 20168806

2. Vũ Hoàng Đức Hiếu MSSV 20168199

3. Nguyễn Hoàng Nhung MSSV 20168399

HN 12/2019

**MỤC LỤC**

[**I.** **ĐỊNH NGHĨA** 4](#_Toc27939027)

[a. Tấn công DoS 4](#_Toc27939028)

[b. Tấn công DDoS 4](#_Toc27939029)

[c. Cấu trúc mạng botnet. 6](#_Toc27939030)

[**II.** **PHÂN LOẠI** 9](#_Toc27939031)

[a. Tấn công vào băng thông mạng – Network-centric Attack 9](#_Toc27939032)

[b. Tấn công vào tài nguyên hệ thống – OS resource Attack 9](#_Toc27939033)

[c. Tấn công vào tài nguyên ứng dụng – Application resource Attack 10](#_Toc27939034)

[**III.** **PHÂN TÍCH VÀ THỬ NGHIỆM MỘT SỐ PHƯƠNG PHÁP TẤN CÔNG DOS TRONG MẠNG** 10](#_Toc27939035)

[a. Ping of Death 10](#_Toc27939036)

[b. Teardrop 14](#_Toc27939037)

[c. TCP SYN Flood 17](#_Toc27939038)

[d. DNS Amplification Attack 20](#_Toc27939039)

[**IV.** **TRIỂN KHAI MỘT SỐ BIỆN PHÁP PHÒNG THỦ** 22](#_Toc27939040)

[a. Ping of Death 22](#_Toc27939041)

[b. Teardrop 23](#_Toc27939042)

[c. TCP SYN Flood 25](#_Toc27939043)

[d. DNS Amplification Attack 26](#_Toc27939044)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 28](#_Toc27939045)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Mô hình tấn công DDoS 5](#_Toc27939054)

[Hình 2: Cấu trúc botnet hình sao (Nguồn: CloudFlare) 6](#_Toc27939055)

[Hình 3: Cấu trúc botnet đa máy chủ (Nguồn: CloudFlare) 7](#_Toc27939056)

[Hình 4: Cấu trúc botnet phân cấp (Nguồn: CloudFlare) 7](#_Toc27939057)

[Hình 5: Cấu trúc botnet peer-to-peer (Nguồn: CloudFlare) 8](#_Toc27939058)

[Hình 6: Kịch bản tấn công Ping of Death (Nguồn: CloudFlare) 11](#_Toc27939059)

[Hình 7: Ping bình thường giữa hai máy 12](#_Toc27939060)

[Hình 8: Bắt gói tin Ping trên Wireshark 12](#_Toc27939061)

[Hình 9: Tấn công Ping of Death sử dụng hping3 13](#_Toc27939062)

[Hình 10: Bắt gói tin Ping of Death trên Wireshark 13](#_Toc27939063)

[Hình 11: Lưu lượng Ping of Death 14](#_Toc27939064)

[Hình 12: Kịch bản tấn công Teardrop (Nguồn: Plixer) 15](#_Toc27939065)

[Hình 13: Phân tích gói tin TCP phân mảnh bình thường 15](#_Toc27939066)

[Hình 14: Tấn công Teardrop sử dụng Scapy 16](#_Toc27939067)

[Hình 15: Phân tích tấn công Teadrop trên Wireshark 17](#_Toc27939068)

[Hình 16: Phân tích giao thức bắt tay ba bước TCP 18](#_Toc27939069)

[Hình 17: Kịch bản tấn công TCP-SYN Flood (Nguồn: CloudFlare) 18](#_Toc27939070)

[Hình 18: Kết quả tấn công TCP-SYN Flood bắt được trên Wireshark 19](#_Toc27939071)

[Hình 19: Đo lưu lượng tấn công TCP-SYN Flood 19](#_Toc27939072)

[Hình 20: Kịch bản tấn công DNS Amplification (Nguồn: CloudFlare) 20](#_Toc27939073)

[Hình 21: Tấn công DNS Amplification 21](#_Toc27939074)

[Hình 22: Phân tích tấn công DNS Amplification 21](#_Toc27939075)

[Hình 23: Triển khai phòng chống Ping of Death 22](#_Toc27939076)

[Hình 24: Thử nghiệm phòng chống Ping of Death 23](#_Toc27939077)

[Hình 25: Triển khai phòng chống Teardrop 24](#_Toc27939078)

[Hình 26: Thử nghiệm phòng chống Teadrop 24](#_Toc27939079)

[Hình 27: Triển khai phòng chống TCP-SYN Flood 25](#_Toc27939080)

[Hình 28: Thử nghiệm phòng chống TCP-SYN Flood 26](#_Toc27939081)

[Hình 29: Triển khai phòng chống DNS Amplification 27](#_Toc27939082)

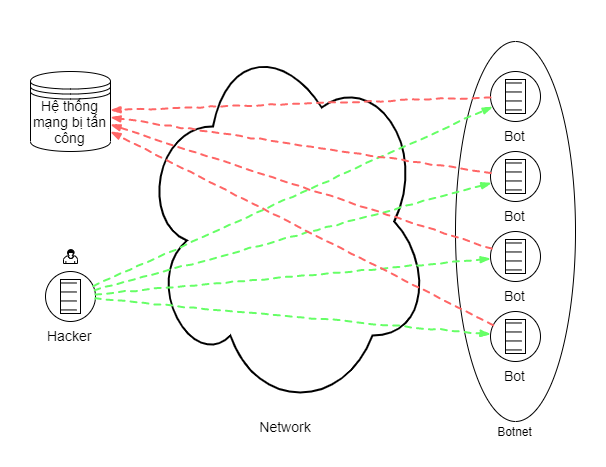
[Hình 30: Thử nghiệm phòng chống DNS Amplification 27](#_Toc27939083)

1. **ĐỊNH NGHĨA**
   1. Tấn công DoS

* Tấn công từ chối dịch vụ, Denial of Service attack, được viết tắt là DoS hoặc DoS attack.
* DoS là một dạng tấn công mạng mà mục tiêu chính là làm cho hệ thống bị tấn công không thể phản hồi lại các yêu cầu truy cập vào một tài nguyên hệ thống.
* Một cuộc tấn công DoS nhằm mục đích ngăn chặn đối với người dùng hợp lệ được cấp quyền hoặc uỷ quyền truy cập vào tài nguyên hệ thống, hoặc trì hoãn các hoạt động, chức năng của hệ thống.
* Kẻ tấn công DoS sẽ nỗ lực để làm cho tài nguyên hệ thống không có sẵn cho người dùng, thông thường mục tiêu sẽ là các máy chủ server web cấu hình cao, nơi cuộc tấn công sẽ khiến cho người dùng không thể truy cập trang web mong muốn.
  1. Tấn công DDoS
* Hệ phân tán Tấn công từ chối dịch vụ, Distributed Denial of Service attack được viết tắt là DDoS hoặc DDoS attack.
* Sự tấn công DoS từ nhiều máy tính hoặc một hệ thống khác với mục tiêu là một hệ thống mạng hoặc server web, gây ra một lưu lượng truy cập lớn hơn sức tải của hệ thống mạng, đến mức làm nghẽn các hoạt động của người dùng hợp lệ được gọi là một cuộc tấn công DDoS
* Kẻ tấn công có thể sử dụng bất kỳ máy tính nào khác bằng việc lợi dụng các lỗ hổng bảo mật hoặc điểm yếu trên máy tính, từ đó chiếm quyền kiểm soát. Sau đó kẻ tấn công có thể bắt máy tính của bạn phải gửi hàng loạt yêu cầu truy cập, hoặc dữ liệu lớn, tới mục tiêu ở đây là một hệ thống mạng hoặc server web khác.
* Cuộc tấn công được thêm từ "Distributed - Phân tán" bởi vì nó không thực hiện từ một máy tính nào mà được "phân tán" ra các máy tính bị chiếm quyền kiểm soát. Mỗi máy tính bị chiếm quyền kiểm soát được gọi là 1 máy bot và hệ thống các máy này được gọi là botnet.

Một vài cuộc tấn công DDoS nổi tiếng:

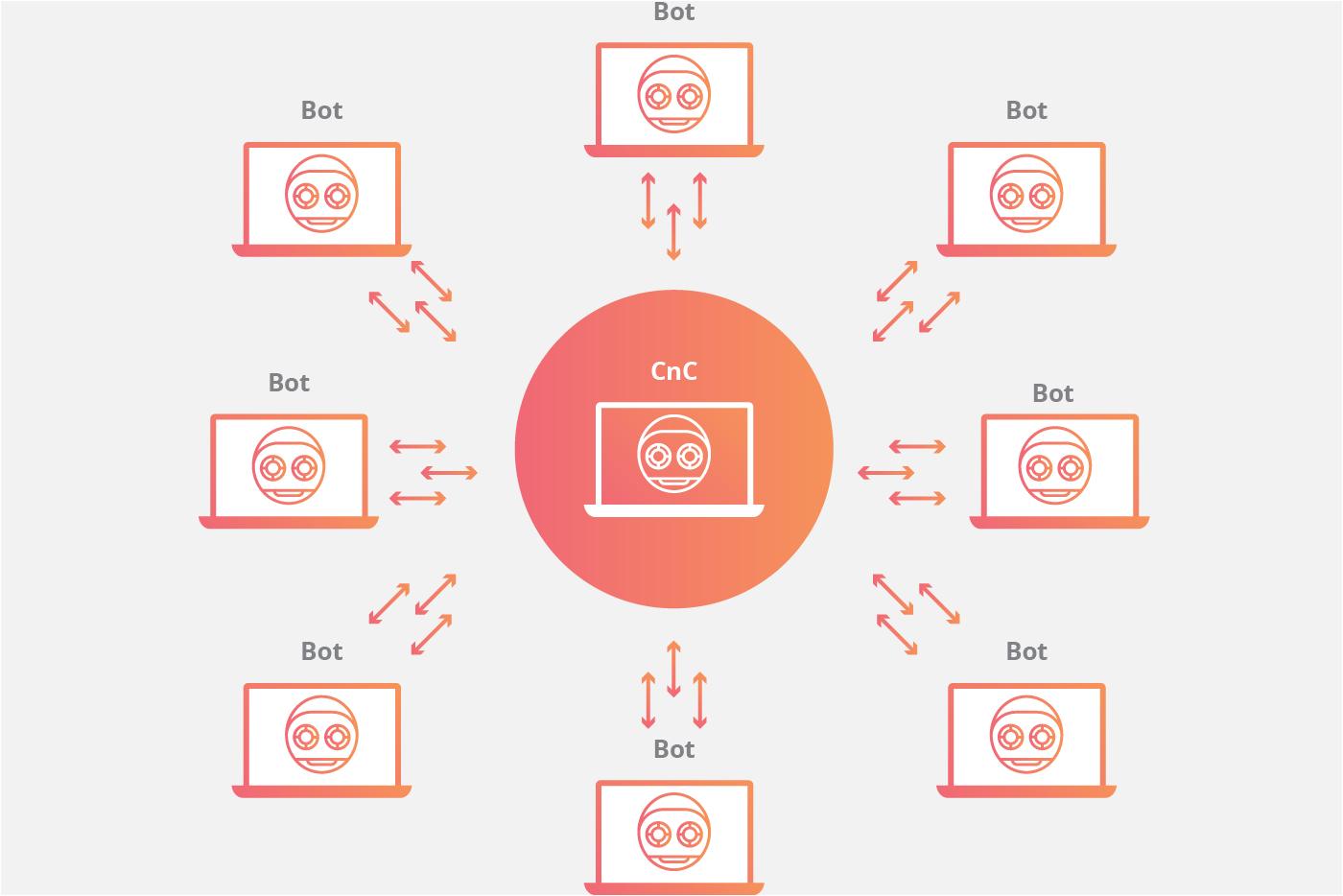
* GITHUB: 1.35 TB/s. Vào ngày 28/2/2018, nền tảng Github - một nền tảng phổ biến để lưu trữ code của lập trình viên - đã bị tấn công với lưu lượng truy cập lên tới 1,35 TB/s.
* Occupy Central Hongkong: 500 GB/s. Vụ tấn công này xảy ra năm 2014, với mục tiêu là hệ thống bỏ phiếu online của cuộc Cách Mạng Dù tại Hồng Kông. Được ghi nhận đỉnh điểm lưu lượng truy cập lên tới 500 GB/s, kéo dài tới hơn 15 phút và lặp lại mỗi vài giờ. Các gói tin được nguỵ trang với lưu lượng hợp lệ được gửi từ không chỉ một hay hai botnet mà năm botnet.
* CloudFlare: 400 GB/s. Cũng vào năm 2014, hệ thống cung cấp dịch vụ bảo mật mạng và phân phối nội dung CloudFlare bị tấn công bởi một kỹ thuật gọi là "reflection - phản xạ". Kẻ tấn công sử dụng một loạt địa chỉ nguồn giả mạo để gửi số lượng lớn các yêu cầu tới máy chủ NTP (Network Time Protocol) và có khả năng phản xạ và khuếch đại gói phản hồi. Một máy tính bị hack với băng thông 1Gbps có thể phản xạ và khuếch đại lên tới 200Gbps.
* United States Banks: 60GB/s. Năm 2012, không chỉ một hay hai mà là cả 6 ngân hàng thuộc hệ thống Ngân hàng Hoa Kỳ đã bị tấn công bởi một chuỗi liên tiếp các cuộc tấn công DDoS. Kẻ tấn công đã sử dụng hàng trăm máy chủ bị nhiễm mã độc để gửi số lượng cực lớn các gói tin gây quá tải, lên tới 60GB/s.



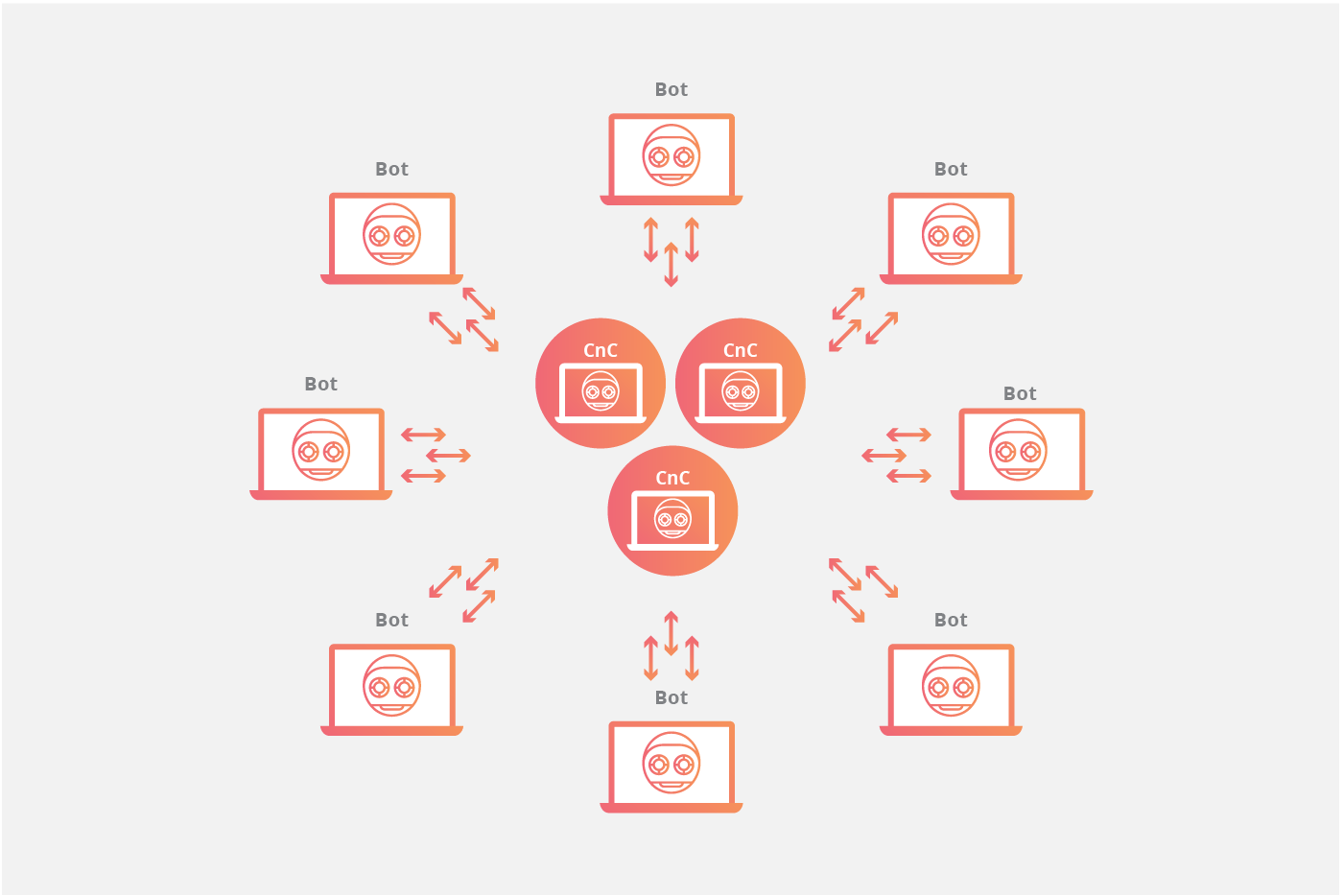
Hình 1: Mô hình tấn công DDoS

* 1. Cấu trúc mạng botnet.
* Botnet là một nhóm các máy tính bị nhiễm mã độc và nằm dưới sự điều khiển từ một phần mềm độc hại. Botnet có thể được thiết kế để dùng vào nhiều mục đích khác nhau, như spam thư rác hoặc tấn công từ chối dịch vụ.
* Botnet có nhiều mức độ khác nhau, một số chiếm hoàn toàn quyền kiểm soát thiết bị, một số lại chạy âm thầm chờ đợi lệnh tấn công từ kẻ tấn công hoặc các máy bot khác. Nó tự lan truyền thông qua các con đường lây nhiễm khác nhau như Trojan, lỗ hổng web, hoặc tự nhân bản cho các thiết bị phần cứng trong cùng mạng nội bộ.
* Đặc điểm cốt lõi của một botnet là khả năng nhận lệnh từ kẻ tấn công hoặc các bot khác. Bên cạnh đó còn là khả năng giao tiếp giữa các bot trong cùng 1 botnet và giữa các botnet với nhau. Thiết kế botnet có thể khác nhau tuỳ vào mục đích sử dụng, nhưng cấu trúc điều khiển có thể chia làm hai loại chính là: cấu trúc máy chủ - máy trạm (client-server) và cấu trúc ngang hàng (peer-to-peer).
* ***Mô hình client-server:***

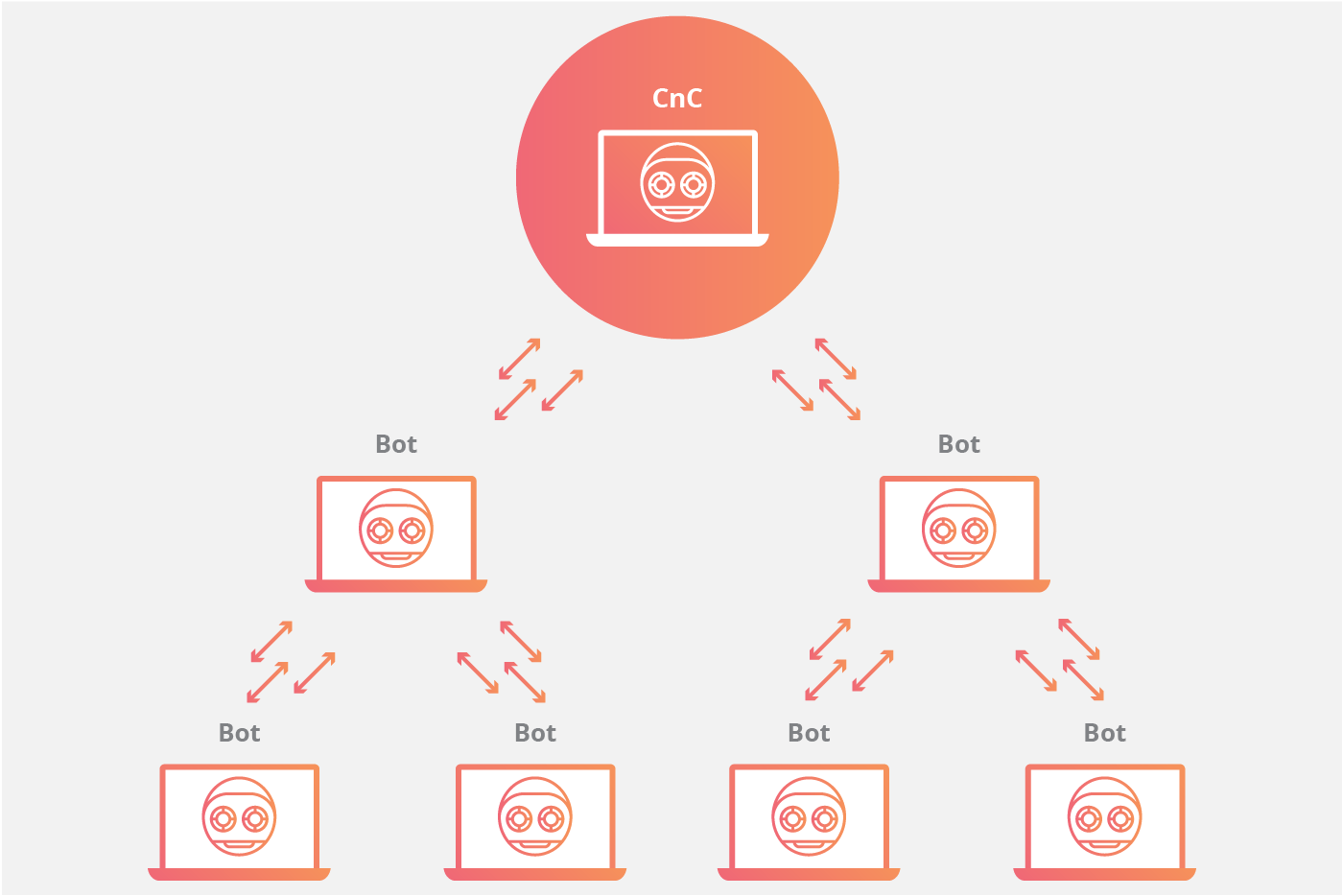
Trong mô hình này, mỗi bot sẽ nhận lệnh từ một hoặc nhiều máy chủ điều khiển (Command and Control center – CnC) để nhận hướng dẫn tấn công. Bằng các sử dụng máy chủ CnC này, kẻ tấn công có thể thay đổi tập lệnh tấn công hoặc mục tiêu tấn công. Máy chủ điều khiển CnC có thể là một botnet hoặc một web server. Một số mô hình cấu trúc liên kết botnet phổ biến: Cấu trúc hình sao (Star Network), Cấu trúc đa máy chủ (Multi Server Network), Cấu trúc phân cấp (Hierarchical Network) …



Hình 2: Cấu trúc botnet hình sao (Nguồn: CloudFlare)

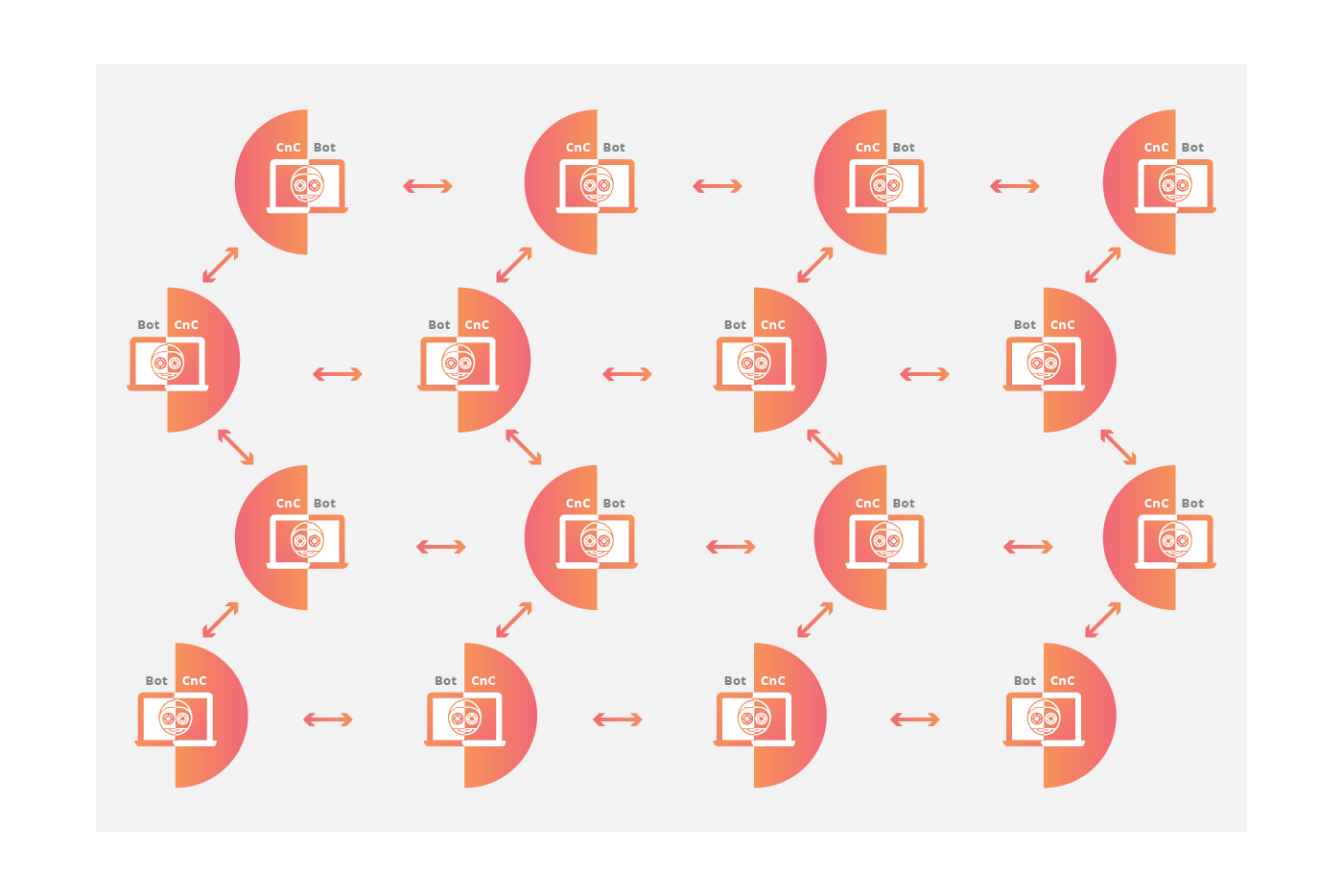


Hình 3: Cấu trúc botnet đa máy chủ (Nguồn: CloudFlare)



Hình 4: Cấu trúc botnet phân cấp (Nguồn: CloudFlare)

* ***Mô hình peer-to-peer:***
* Điểm yếu của mạng botnet có cấu trúc client-server là một khi đánh chặn được máy chủ điều khiển CnC thì toàn bộ mạng botnet sẽ sập theo. Để khắc phục điểm yếu này, các botnet gần đây đã được thiết kế theo mô hình ngang hàng peer-to-peer. Việc nhúng cấu trúc điều khiển này đã loại bỏ điểm yếu phụ thuộc máy chủ điều khiển, khiến cho những nỗ lực giảm thiểu trở nên khó khăn.
* Các botnet ngang hàng duy trì một danh sách các máy tính đáng tin cậy mà từ đó chúng nhận và gửi thông tin điều khiển. Nhưng cấu trúc này cũng có điểm yếu, đó chính là việc thiếu máy chủ điều khiển CnC khiến cho mạng ngang hàng dễ bị chiếm quyền kiểm soát bởi một người khác không phải là người tạo ra botnet.



Hình 5: Cấu trúc botnet peer-to-peer (Nguồn: CloudFlare)

1. **PHÂN LOẠI**

Có nhiều phương pháp để phân loại các dạng tấn công DDoS, ví dụ như:

* Phân loại theo nạn nhân: tấn công vào người dùng khách, tấn công vào máy chủ, tấn công vào hệ thống mạng …
* Phân loại theo mức độ tự động: tấn công thủ công, tấn công bán tự động, tấn công tự động….
* Phân loại theo cách thức tấn công: tấn công trực diện, tấn công ánh xạ, …

Trên phạm vi bài báo cáo nhóm xin được phân loại dựa theo mục tiêu tấn công. Có nhiều mục tiêu khi thực hiện tấn công DDoS, nhưng tập trung chủ yếu vào 3 mục tiêu chính: băng thông mạng, tài nguyên hệ thống và tài nguyên ứng dụng.

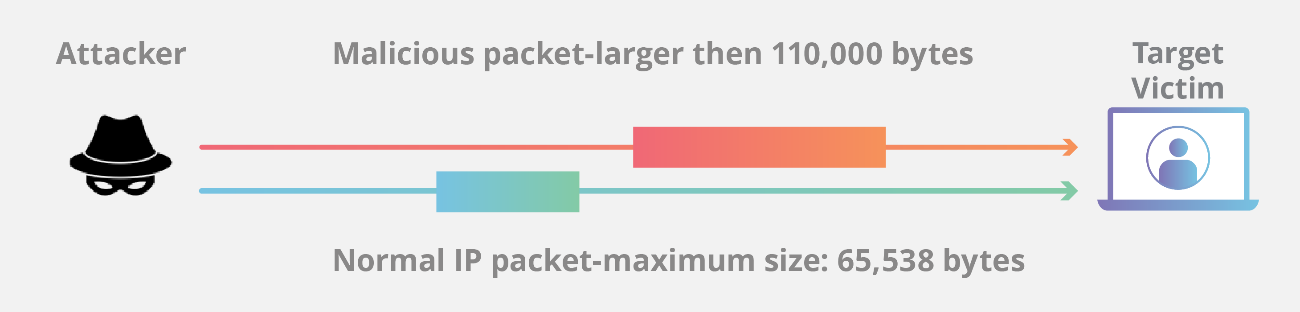
* 1. Tấn công vào băng thông mạng – Network-centric Attack
* Mục đích chính của lại tấn công này là gây ra tắc nghẽn trên đường truyền mạng, tiêu thụ toàn bộ băng thông có sẵn giữa mục tiêu và Internet. Một lượng lớn dữ liệu được gửi đến mục tiêu bằng cách sử dụng hình thức khuếch đại hoặc một phương tiện khác, chẳng hạn như các yêu cầu từ botnet. Thông thường thì băng thông được cung cấp bởi nhà cung cấp dịch vụ mạng (ISP) thường lớn hơn băng thông của máy chủ, nên sẽ dẫn tới tình trạng không đáp ứng được cho người dùng hợp lệ.
* Trong các kiểu tấn công vào băng thông, thì phổ biến nhất là tấn công gây lụt băng thông mạng. Một số kiểu tấn công thường gặp: ICMP Flood, UDP Flood, HTTP Flood, … Kiểu tấn công này thường được đo bằng đơn vị bit per second, bps hoặc Gbps.
  1. Tấn công vào tài nguyên hệ thống – OS resource Attack
* Mục đích chính của loại tấn công này là tạo một lượng lớn dữ liệu cần xử lý tại máy chủ, hoặc gây lỗi cho máy chủ khi xử lí các gói tin. Nó gây tiêu tốn tài nguyên máy chủ (CPU, RAM, SSD, …) hoặc các thiết bị giao tiếp trung gian khác như tường lửa (firewall) hoặc bộ cân bằng tải (load balancers)
* Thường gặp nhất ở kiểu tấn công này là TCP-SYN Flood, Ping of Death, Smurf DDoS … Để đo lưu lượng khi bị tấn công vào tài nguyên hệ thống, ta dùng đơn vị package per second, pps.

* 1. Tấn công vào tài nguyên ứng dụng – Application resource Attack
* Một máy chủ khi hoạt động sẽ được sử dụng cho mục đích cuối cùng là đáp ứng nhu cầu của người dùng ứng dụng. Máy chủ ở đây có thể là máy chủ web-server, máy chủ cơ sở dữ liệu, máy chủ game-server, …Kiểu tấn công vào tài nguyên ứng dụng này nhằm mục đích tiêu tốn toàn bộ tài nguyên của ứng dụng đang hoạt động tại server, khiến cho các yêu cầu hợp lệ bị từ chối hoặc chậm trễ. Tấn công vào tầng ứng dụng rất khó để đề phòng vì lưu lượng giả nhận được gần như không phân biệt được hoặc rất khó phân biệt với lưu lượng của người dùng hợp lệ.
* Điển hình nhất cho kiểu tấn công này chính là HTTP Flood, một kiểu tấn công trực tiếp vào cả tài nguyên ứng dụng và băng thông mạng. Bên cạnh đó còn có một số kiểu tấn công khác như DNS Amplification Attack, NTP Amplification Attack, … Với hình thức tấn công này, đơn vị đo lưu lượng tấn công là request per second, rps.

1. **PHÂN TÍCH VÀ THỬ NGHIỆM MỘT SỐ PHƯƠNG PHÁP TẤN CÔNG DOS TRONG MẠNG**

Trong phạm vi bài báo cáo, nhóm xin trình bày về một số phương pháp tấn công và phòng chống cơ bản của một số phương thức tấn công DoS phổ biến trong mạng, đó là Ping of Death, Teardrop, TCP SYN Flood, DNS Amplification Attack. Các công cụ sử dụng bao gồm:

* Phần mềm ảo hoá VirtualBox 6.0.14
* Hệ điều hành Ubuntu 18.04, Window 10 Pro 1909.
* Các phần mềm, công cụ khác: Wireshark, hping3, Python Scapy, netcat, tcpdump, dnsdrdos.c, …
  1. Ping of Death
* Ping là một phương pháp để kiểm tra kết nối giữa hai máy, sử dụng giao thức ICMP (Internet Control Message Protocol). Máy khách sẽ gửi một gói tin ICMP đến máy chủ với trường Type = 8 (Ping Request) và chờ một gói tin ICMP phản hồi lại với trường Type = 0 (Ping Reply). Gói tin ICMP có độ dài tối đa là 576 bytes (theo RFC1812). Dữ liệu được đóng gói lại thành 1 gói tin IP, có độ dài tối đa là 65535 bytes với phần header là gói tin ICMP và phần dữ liệu gửi đi và được gửi ngược lại trong gói tin Ping Reply để xác nhận.
* Sử dụng công cụ hping3, ta có thể khiến cho gói dữ liệu được gửi kèm này vượt quá tiêu chuẩn của IPv4 là 65535 bytes. Giao thức IPv4 sẽ phân mảnh gói tin thành các gói tin có độ dài 1500 bytes với 20 bytes header và 1480 bytes data. Việc gửi gói tin lớn hơn 65535 bytes sẽ khiến máy chủ khi nhận được hàng loạt gói tin và không thể tái hợp lại thành gói tin ban đầu, gây lỗi hệ thống.
* Ping of Death thuộc dạng tấn công vào tài nguyên hệ thống, thường được kết hợp với ICMP Flood (gửi đồng loạt nhiều gói tin ping) để tấn công cả vào băng thông mạng.
* Kịch bản tấn công:



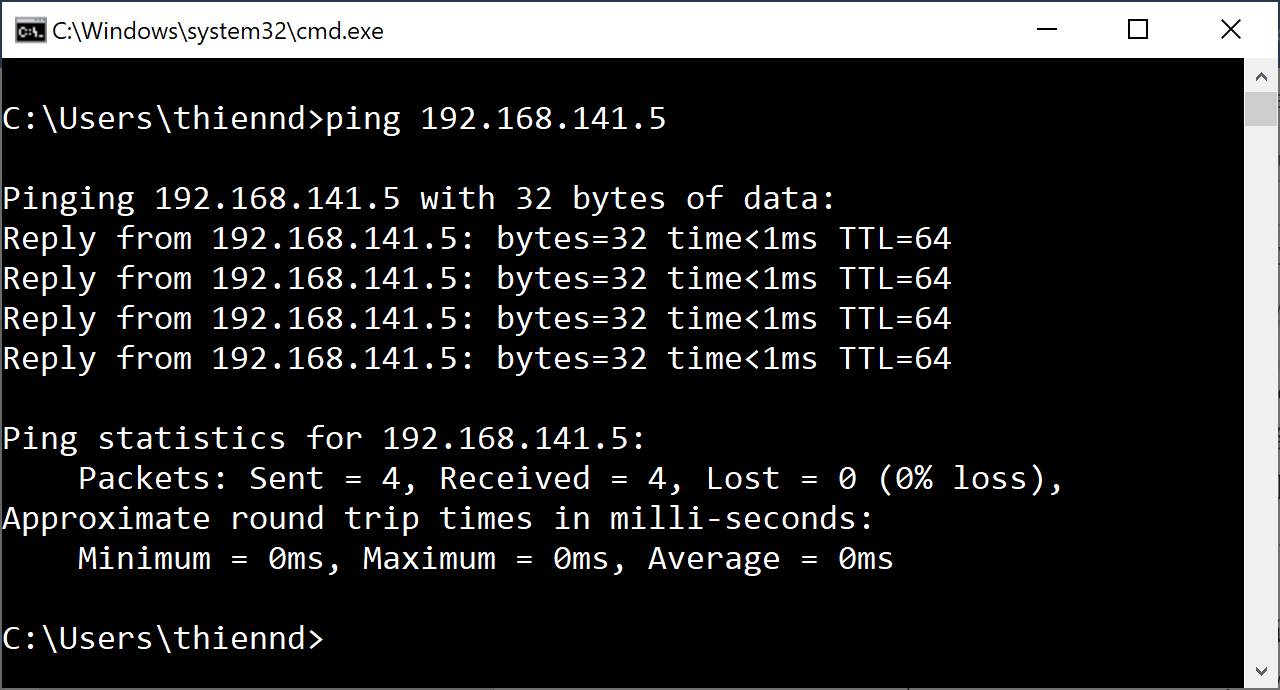
Hình 6: Kịch bản tấn công Ping of Death (Nguồn: CloudFlare)

* Các hệ điều hành tiêu biểu bị ảnh hưởng bởi Ping of Death:
  + Solaris (x86) phiên bản < 2.4: khiến hệ thống bị khởi động lại.
  + HP 3000 MPE/iX phiên bản 4.0, 5.0, 5.5: khiến hệ thống từ chối các gói tin với thông báo lỗi “System abort 3890 from subsys 201”.
  + Apple Mac OS phiên bản < 7.x.x:gây sập máy chủ.
  + Window 95 tất cả các phiên bản: gây sập hệ thống
  + Linux phiên bản <= 2.0.23: gây khởi động lại hoặc lỗi kernel panic
* Thử nghiệm:

Máy ảo Ubuntu đã được cài đặt hping3.

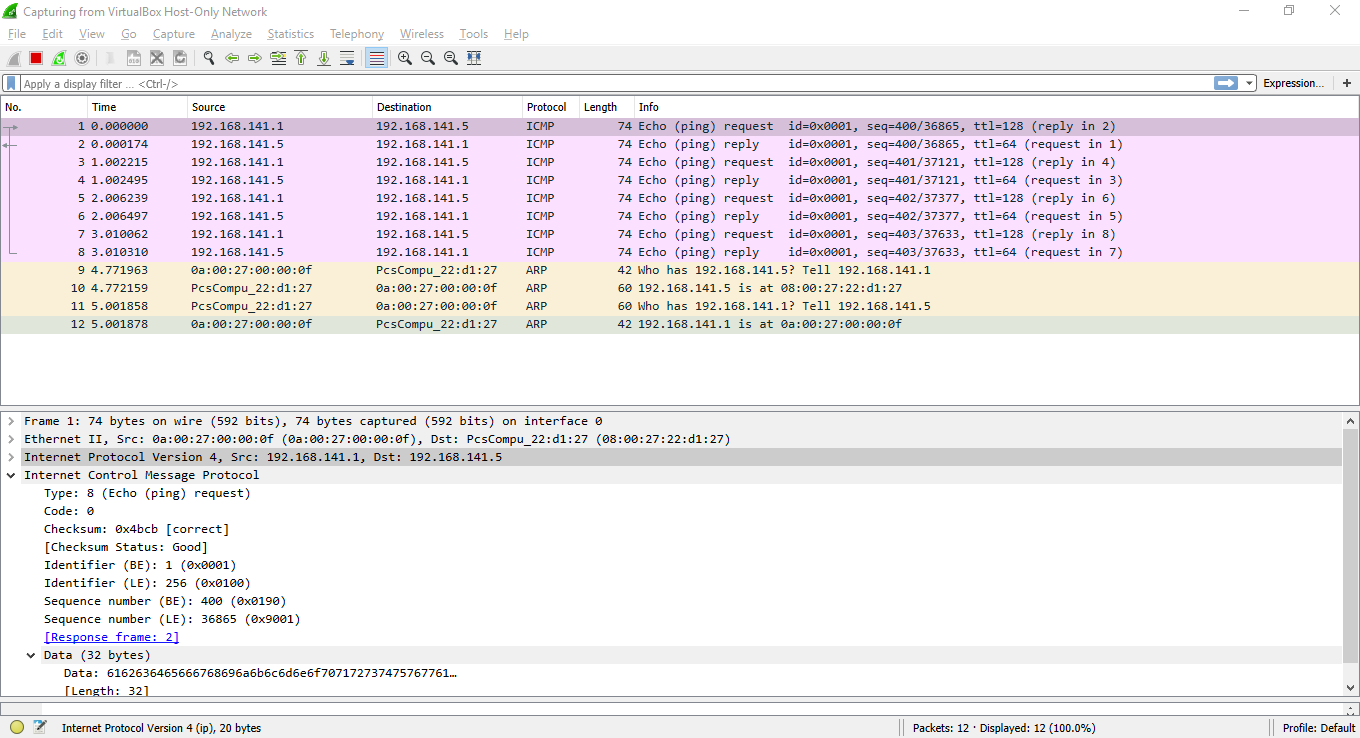
Máy chủ Window đã cài đặt Wireshark.

* Thực hiện ping bình thường để kiểm tra kết nối giữa hai máy:



Hình 7: Ping bình thường giữa hai máy

Gói tin bắt được trên Wireshark:

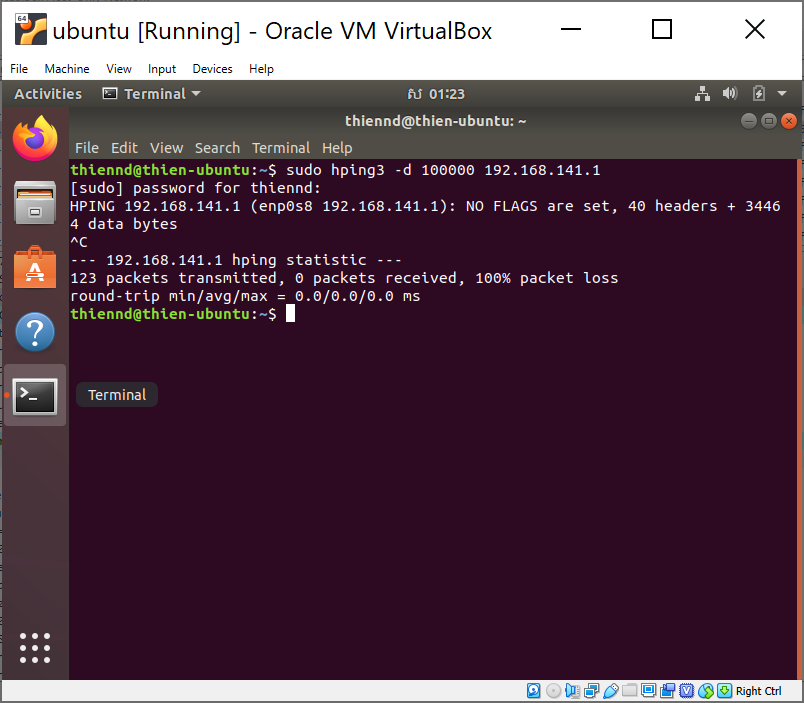


Hình 8: Bắt gói tin Ping trên Wireshark

Lệnh ping thực hiện bình thường, gửi đi 32 bytes dữ liệu và nhận về 32 bytes dữ liệu. Bắt được các gói tin Ping Request và Ping Reply chuẩn trong Wireshark.

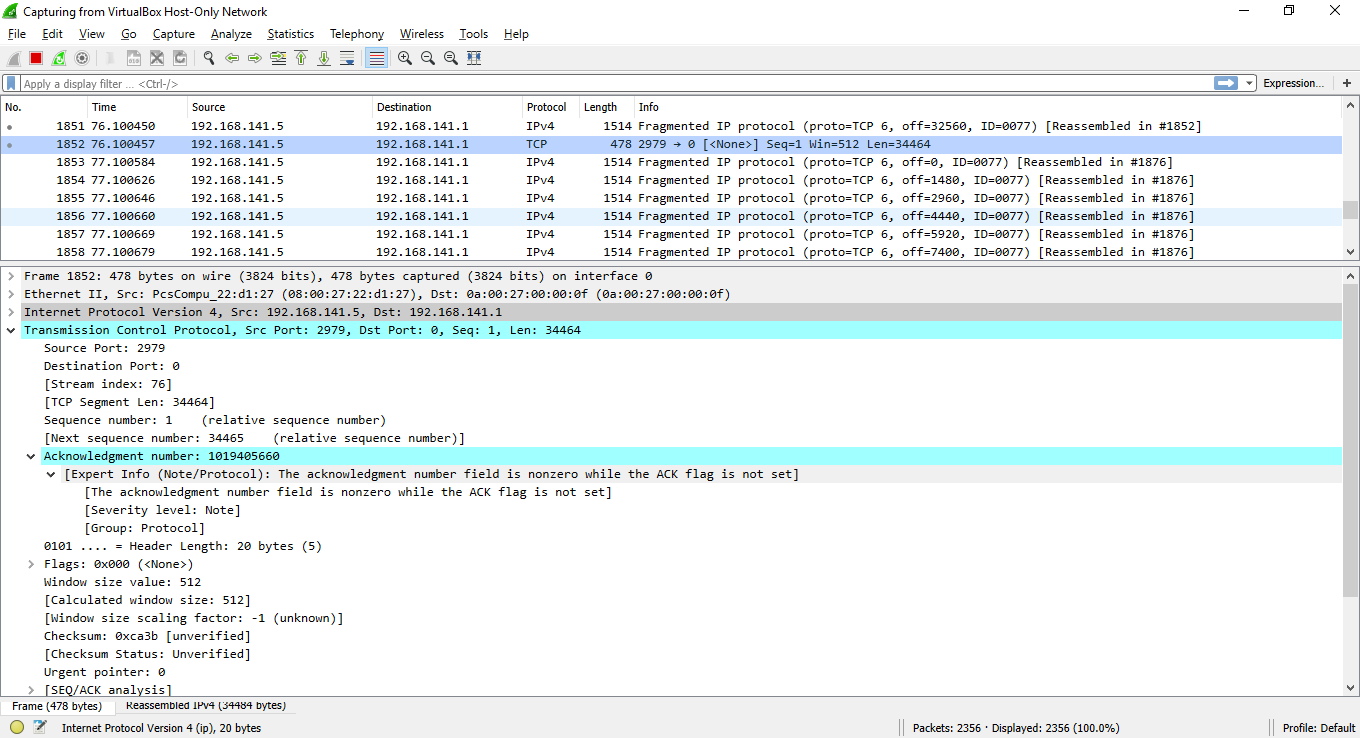
* Thực hiện tấn công từ máy Ubuntu với lệnh sau:

**sudo hping3 -d 100000 192.168.141.1**



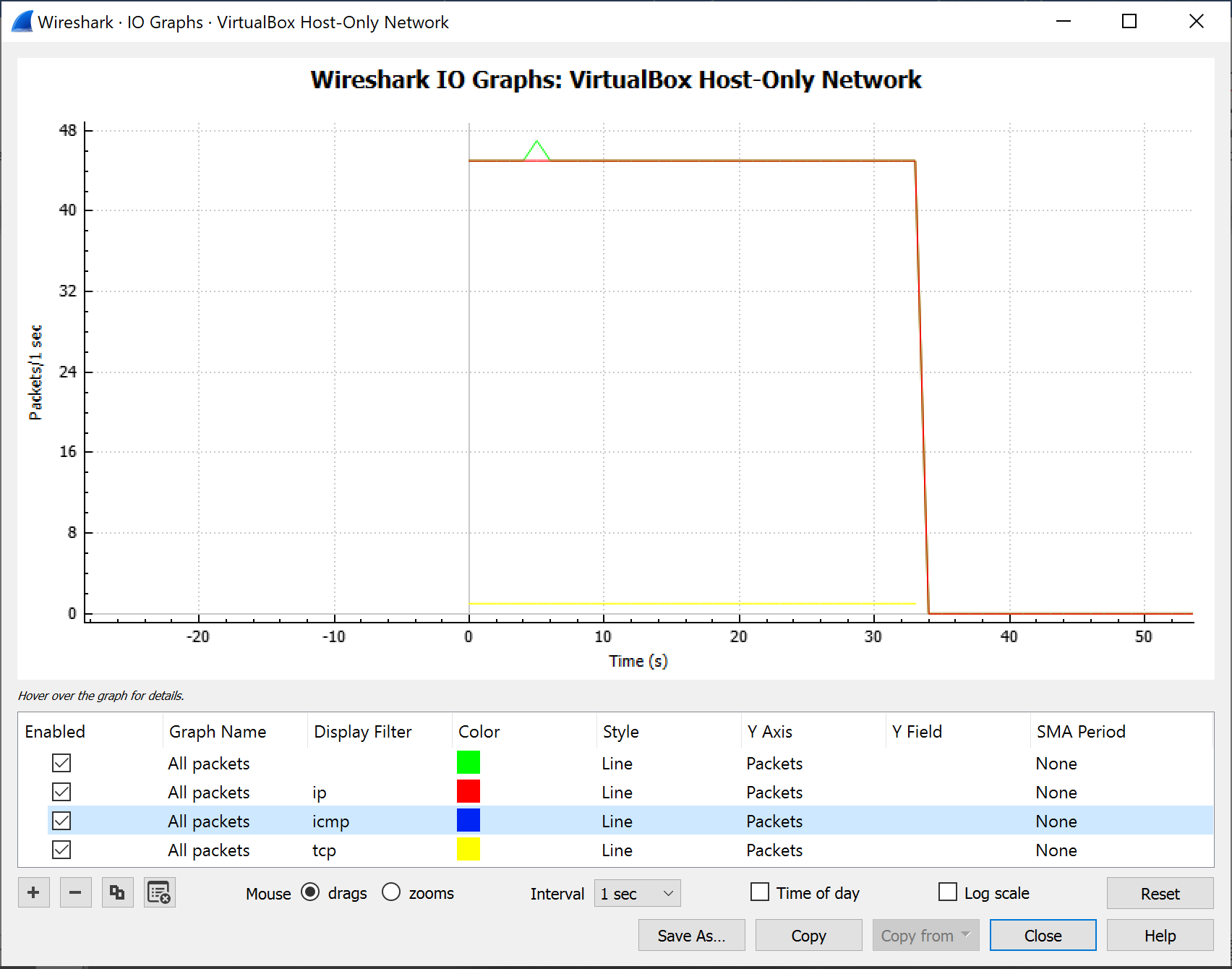
Hình 9: Tấn công Ping of Death sử dụng hping3

* Gói tin bắt được trên Wireshark:



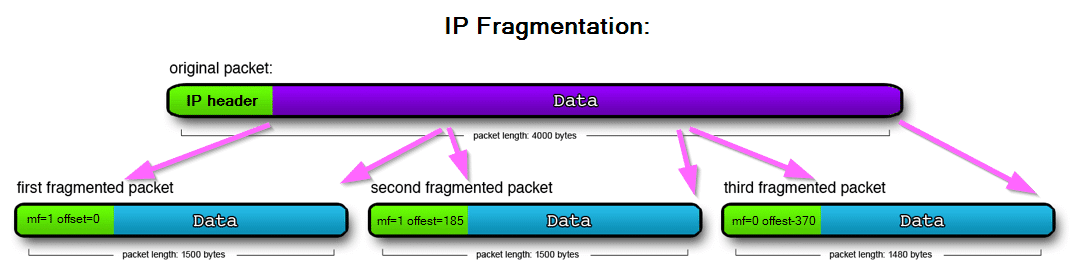
Hình 10: Bắt gói tin Ping of Death trên Wireshark

* Gói tin ping được gửi đi không đơn giản là ICMP nữa mà được đóng gói thành IPv4 với dữ liệu phân mảnh gửi kèm. Do dữ liệu gửi kèm đã vượt quá 65535 bytes nên máy chủ không thể phản hồi lại gói tin ban đầu. Kết quả thu được trên Ubuntu là 123 packets đã gửi, 0 packets đã nhận.
* Lưu lượng đo được trên máy Window bằng công cụ Statistics – I/O Graph của Wireshark: 47 pps



Hình 11: Lưu lượng Ping of Death

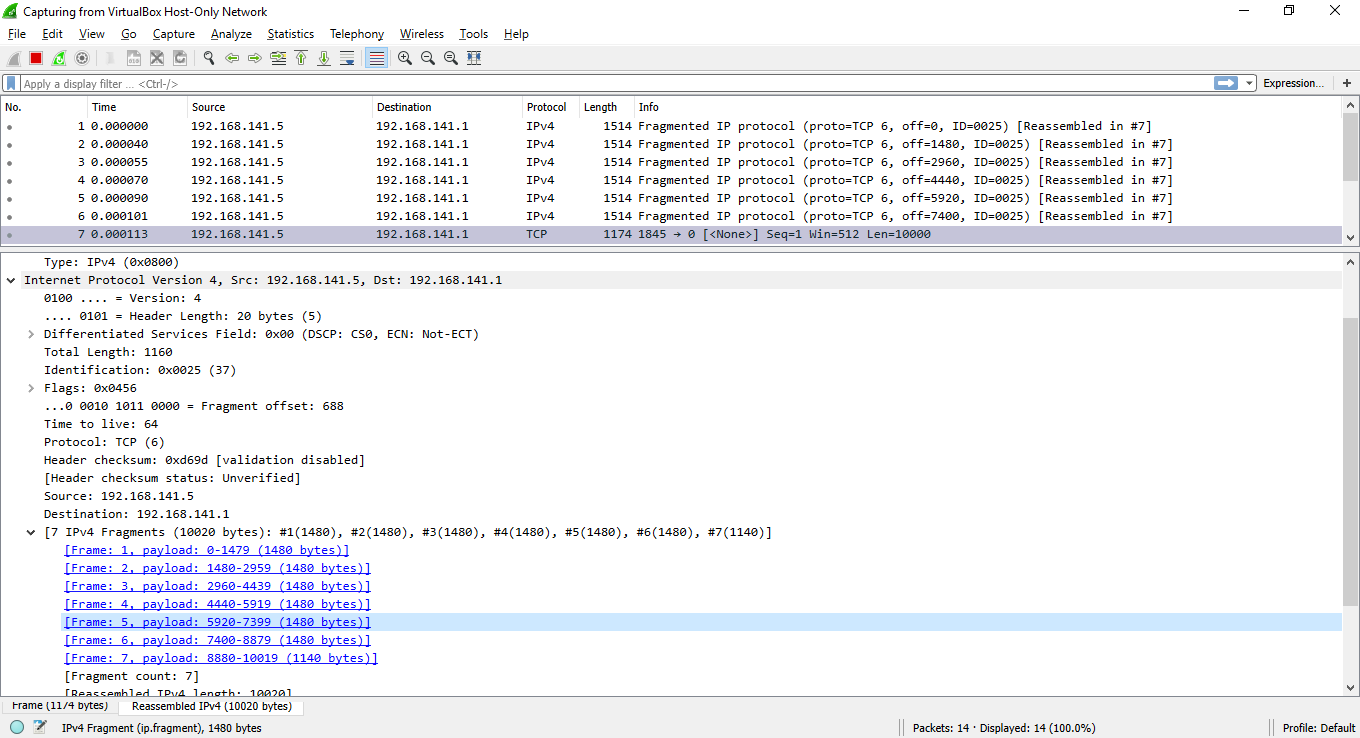
* 1. Teardrop
* Giao thức TCP được dùng để điều khiển luồng dữ liệu tại tầng giao vận trong kiến trúc OSI. Dữ liệu được chia ra và gửi đi dưới dạng các mảnh dữ liệu, được đóng gói trong gói tin IPv4 với kích thước theo kích thước tối đa (MTU – Maximum Transmission Unit) mà tầng liên kết cho phép, với liên kết Ethernet thì MTU=1500. Trong đó có 20 bytes là header của gói tin, còn lại 1480 bytes là dữ liệu được gửi. Sau đó máy chủ nhận được sẽ ghép các mảnh dữ liệu thành gói tin TCP hoàn chỉnh bằng các số offset được ghi trong header.
* Tấn công Teardrop tập trung vào việc sửa số offset này, dẫn đến máy chủ không thể ghép các mảnh lại một cách chính xác thành gói dữ liệu ban đầu. Từ đó dẫn đến các gói tin TCP tiếp sau cũng không thể ghép lại được. Tấn công Teardrop hay còn được gọi là Fragmentation TCP Attack – tấn công phân mảnh giao thức TCP, một dạng của tấn công phân mảnh Fragmentation Attack. Tác hại của tấn công Teardrop là gây lỗi bộ nhớ máy tính, thuộc kiểu tấn công với mục tiêu là gây lỗi tài nguyên hệ thống máy nạn nhân.
* Kịch bản tấn công:
* Các gói tin được chia nhỏ nhưng giá trị offset bị sai. Gói tin có độ dài 4000 bytes nếu đúng ra sẽ được chia thành 3 gói tin với MTU=1500, offset sẽ lần lượt là 0, 1480, 2960. Nhưng ở đây offset đã bị chỉnh sửa thành 0, 185, 370.



Hình 12: Kịch bản tấn công Teardrop (Nguồn: Plixer)

* Thử nghiệm:
* Máy chủ Window cài đặt Wireshark
* Máy tấn công Ubuntu cài đặt Python Scapy

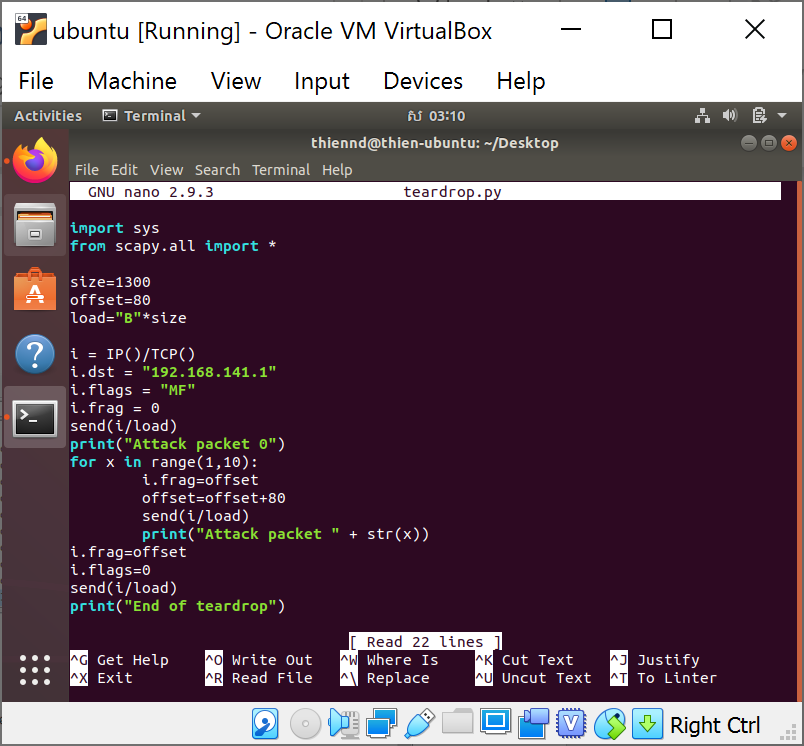
Gói tin TCP bình thường:



Hình 13: Phân tích gói tin TCP phân mảnh bình thường

Các gói tin IPv4 từ 1 – 6 có offset đúng theo thứ tự 0-1479, 1480-2960, … và được ghép lại hoàn chỉnh tại gói tin TCP .

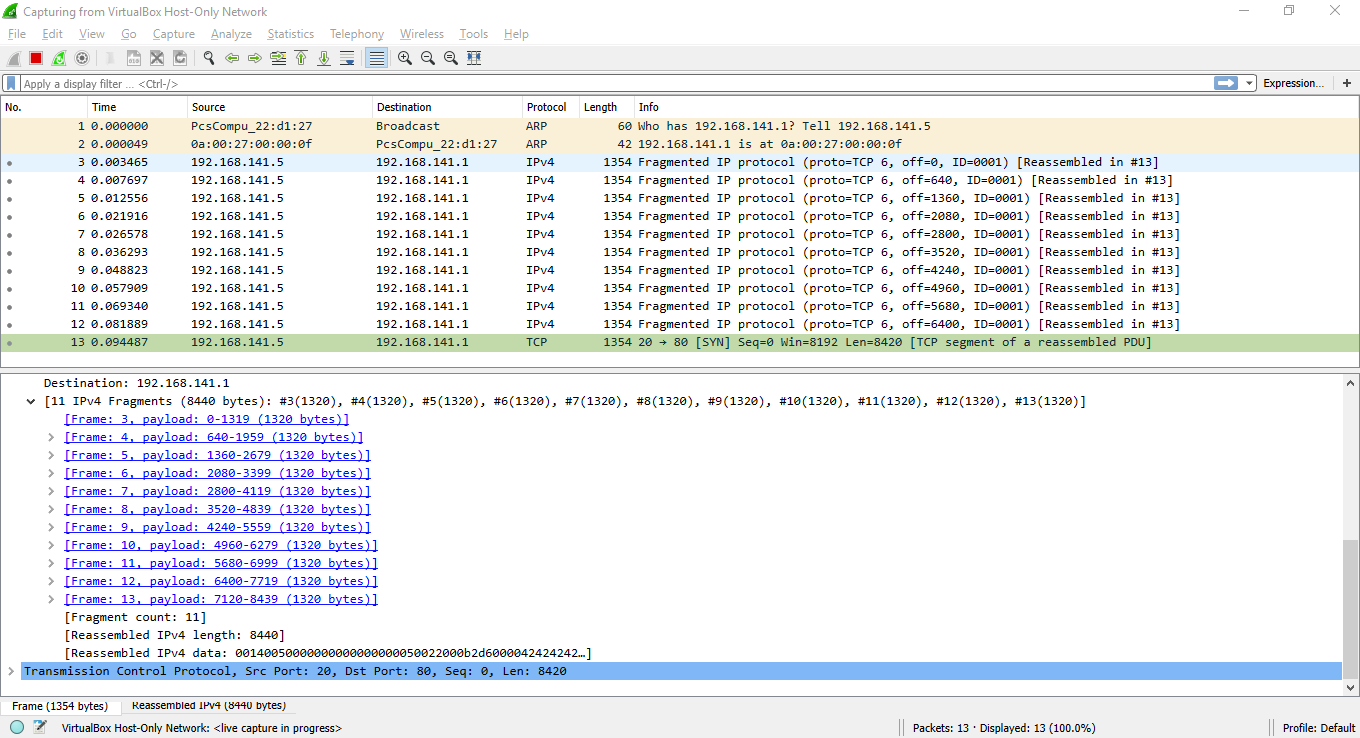
Sử dụng thư viện Scapy để gửi 10 gói tin TCP với độ dài 1300 bytes, và offset không chính xác.



Hình 14: Tấn công Teardrop sử dụng Scapy

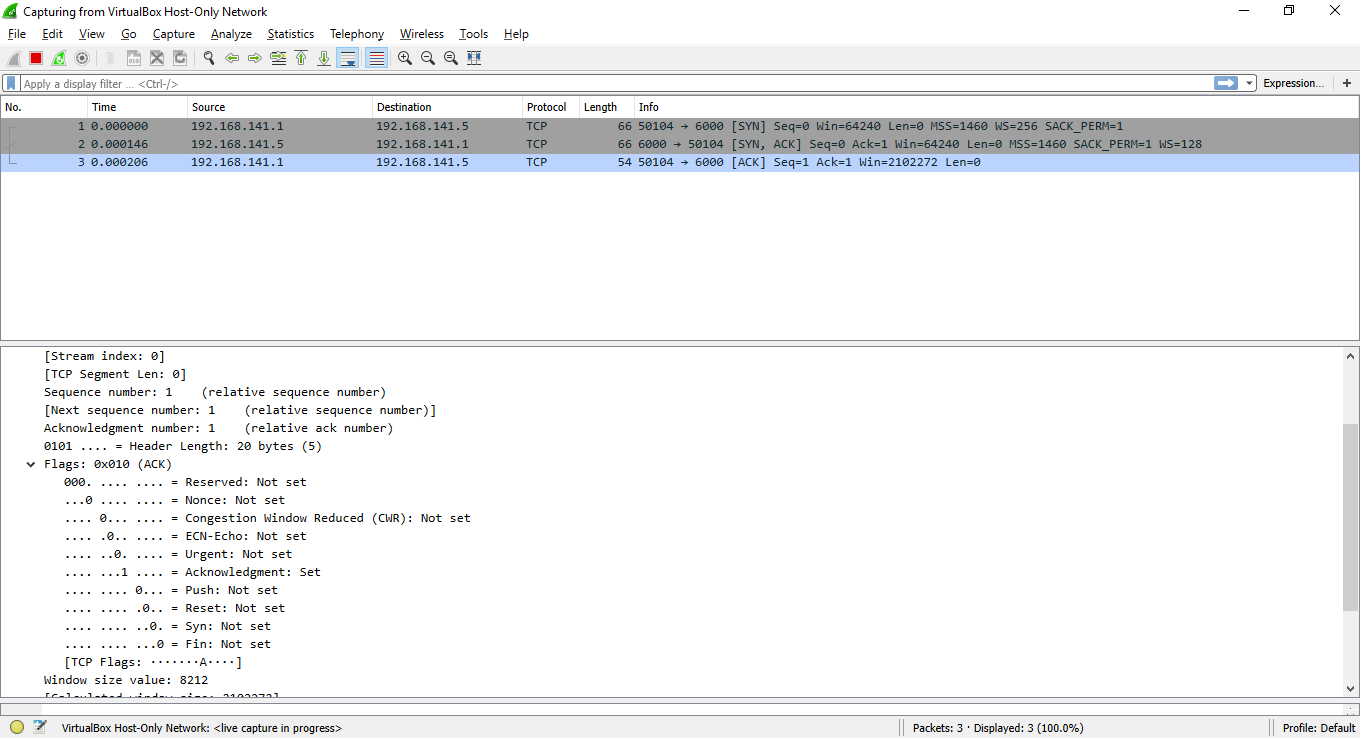
* Đầu tiên ta gửi 1 gói tin TCP/IP với header có flags được gán giá trị MF = More Fragment, offset=0, data=1300bytes. Với cờ MF thì máy nạn nhân sẽ tiếp tục chờ để nhận thêm dữ liệu.
* Lúc này ta gửi thêm 10 gói tin nữa với kích thước data=1300 bytes nhưng giá trị offset chỉ tăng lên 80 sau mỗi lần gửi.
* Cuối cùng ta gửi 1 gói tin có flags = 0 để thông báo kết thúc truyền tin. Lúc này máy nạn nhân sẽ không thể ghép các mảnh tin IPv4 lại thành 1 gói tin TCP/IP hoàn chỉnh được do thông số offset bị sai.

Kết quả thu được tại Wireshark:



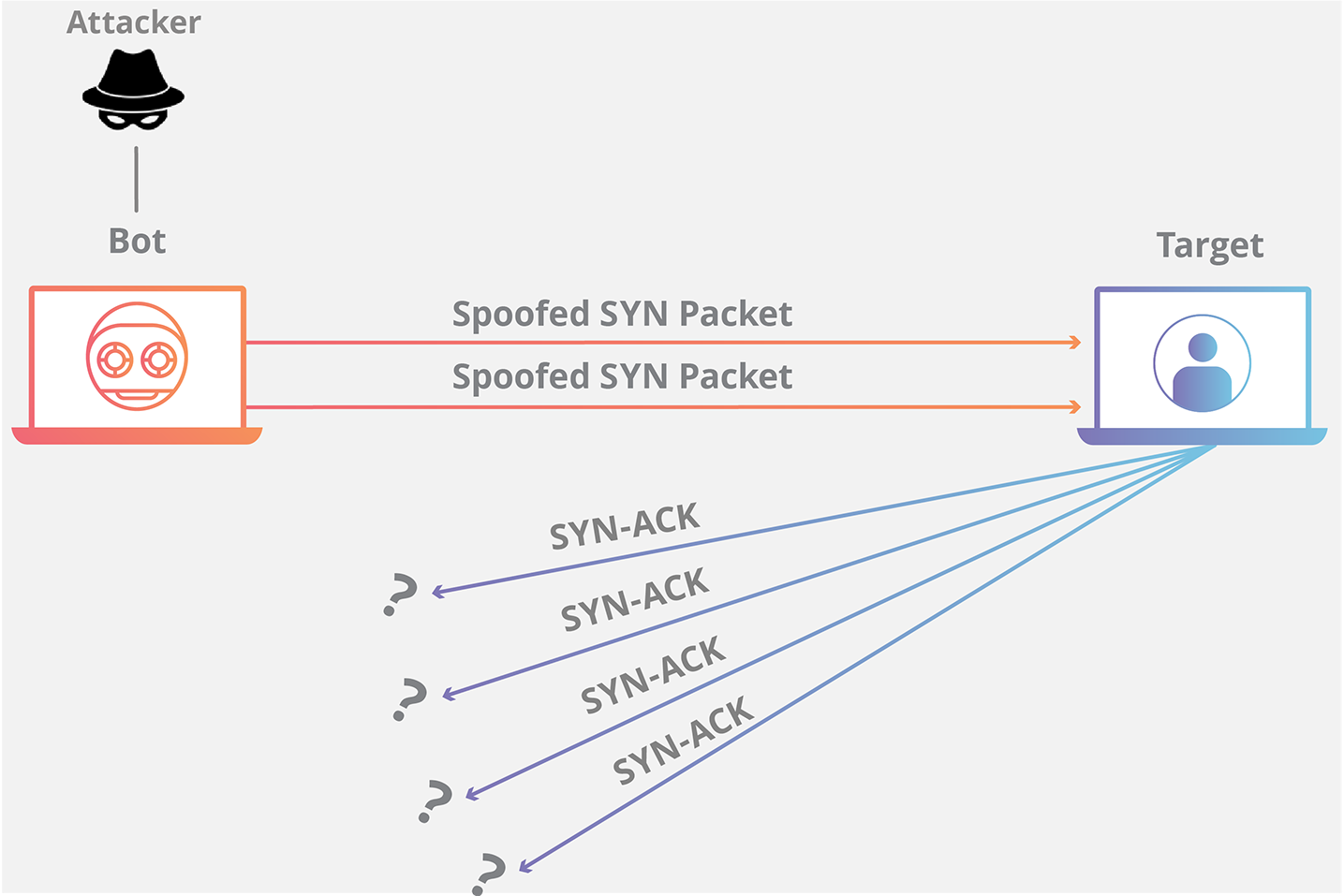
Hình 15: Phân tích tấn công Teadrop trên Wireshark

* 1. TCP SYN Flood
* Giao thức TCP là một giao thức hướng kết nối, yêu cầu hai bên xác thực bắt tay 3 bước trước khi truyền dữ liệu, không giống với UDP là giao thức có thể gửi gói tin đi mà không cần thiết lập kết nối. Việc thiết lập kết nối được thực hiện theo phương thức bắt tay 3 bước như sau:
* Client gửi yêu cầu kết nối với Server, gửi gói tin TCP với cờ SYN được bật, giá trị flag trong header gói tin TCP là 0x002. Trong gói tin này được gửi kèm tham số sequence number với giá trị X ngẫu nhiên.
* Server đáp lại bằng gói tin TCP với cờ SYN và ACK cùng được bật, giá trị flag trong header gói tin TCP lúc này là 0x012. Trong gói tin này tham số sequence number là một giá trị Y ngẫu nhiên, còn giá trị acknowledgement number là giá trị X+1 với X từ gói tin TCP SYN.
* Client nhận được gói tin SYN-ACK và hồi đáp lại bằng gói tin TCP với cờ ACK được bật, flag = 0x010. Trong gói tin này là tham số ACK có giá trị bằng Y+1.
* Sử dụng công cụ netcat ta dễ dàng mô phỏng lại được quá trình này bằng các lệnh sau:
* Máy Ubuntu: Bật chế độ lắng nghe tại cổng 6000: **nc -l -p 6000**
* Máy Window: Kết nối tới cổng 6000: **nc 192.168.141.5 6000**
* Lúc này ta thu được kết quả trên Wireshark lần lượt là 3 gói tin TCP-SYN, SYN-ACK, TCP-ACK



Hình 16: Phân tích giao thức bắt tay ba bước TCP

* Giao thức trên có điểm yếu là tại sau bước thứ 2, server sẽ chờ gói tin ACK phản hồi lại từ client, và nếu client này là giả mạo hoặc không có gói tin ACK phản hồi lại thì server sẽ chờ cho tới khi bị lỗi request time out. Điều này gây tốn tài nguyên máy chủ. Và nếu một lượng lớn gói tin TCP-SYN gửi đến mà không phản hồi lại gói tin TCP-ACK thì có thể khiến máy chủ bị trì trệ các gói tin TCP-SYN khác. Đây chính là kiểu tấn công TCP-SYN Flood.
* Kịch bản tấn công:
* Gửi hàng loạt gói tin yêu cầu kết nối TCP với giá trị IP Source giả mạo, cờ SYN được bật.
* Server nhận được gói tin TCP-SYN, sẽ gửi lại gói tin SYN-ACK tới địa chỉ IP giả mạo và chờ nhận lại gói tin TCP-ACK.

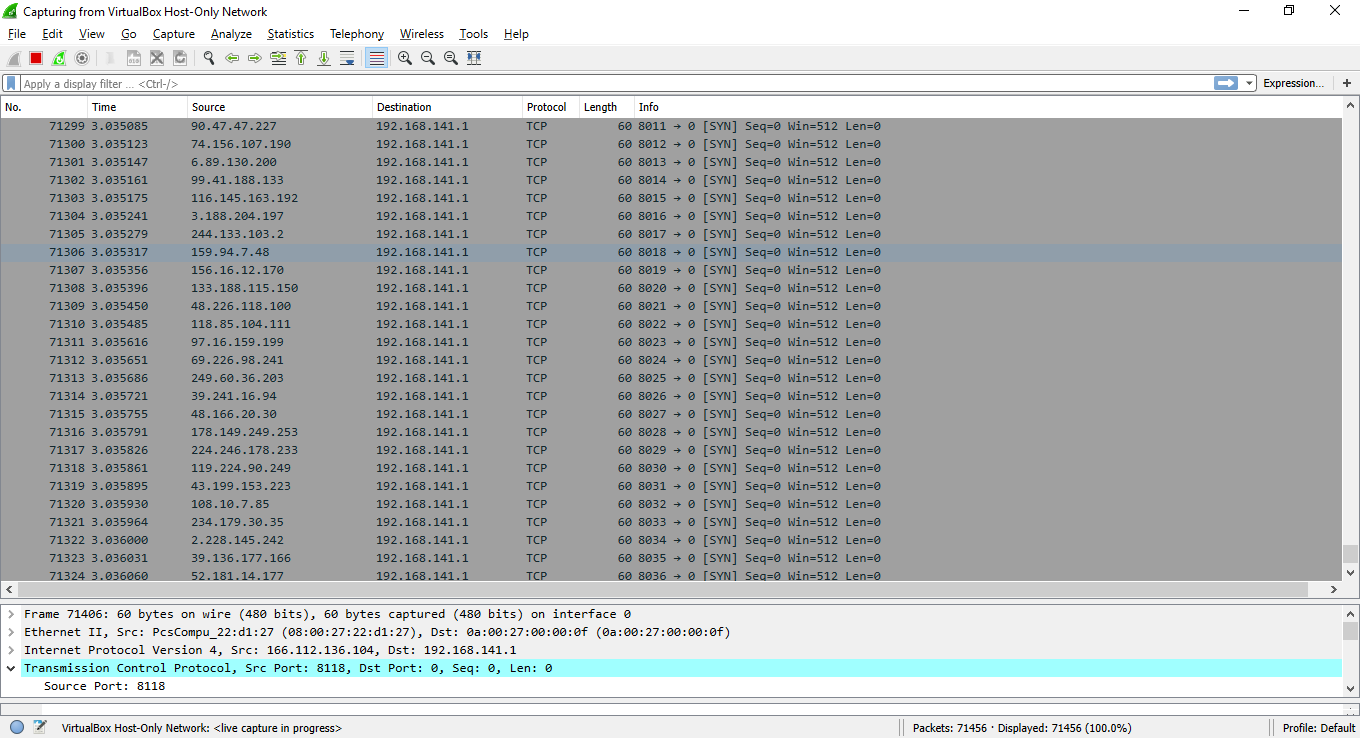


Hình 17: Kịch bản tấn công TCP-SYN Flood (Nguồn: CloudFlare)

* Thử nghiệm tấn công:
* Máy Window cài Wireshark để bắt gói tin
* Máy Ubuntu cài hping3 để tấn công

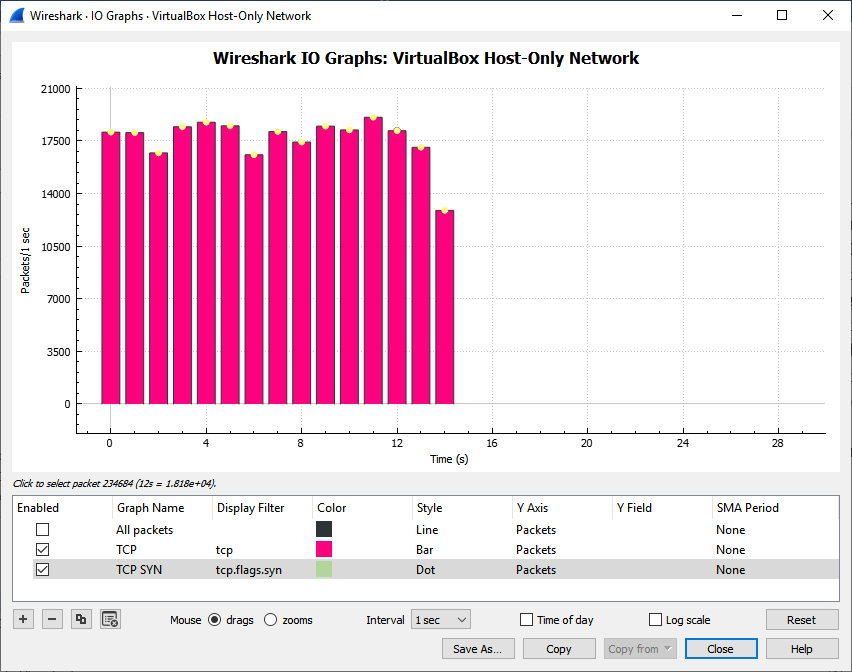
Câu lệnh tấn công: **sudo hping3 -S --flood --rand-source 192.168.141.1**

Thu được kết quả trên Wireshark:



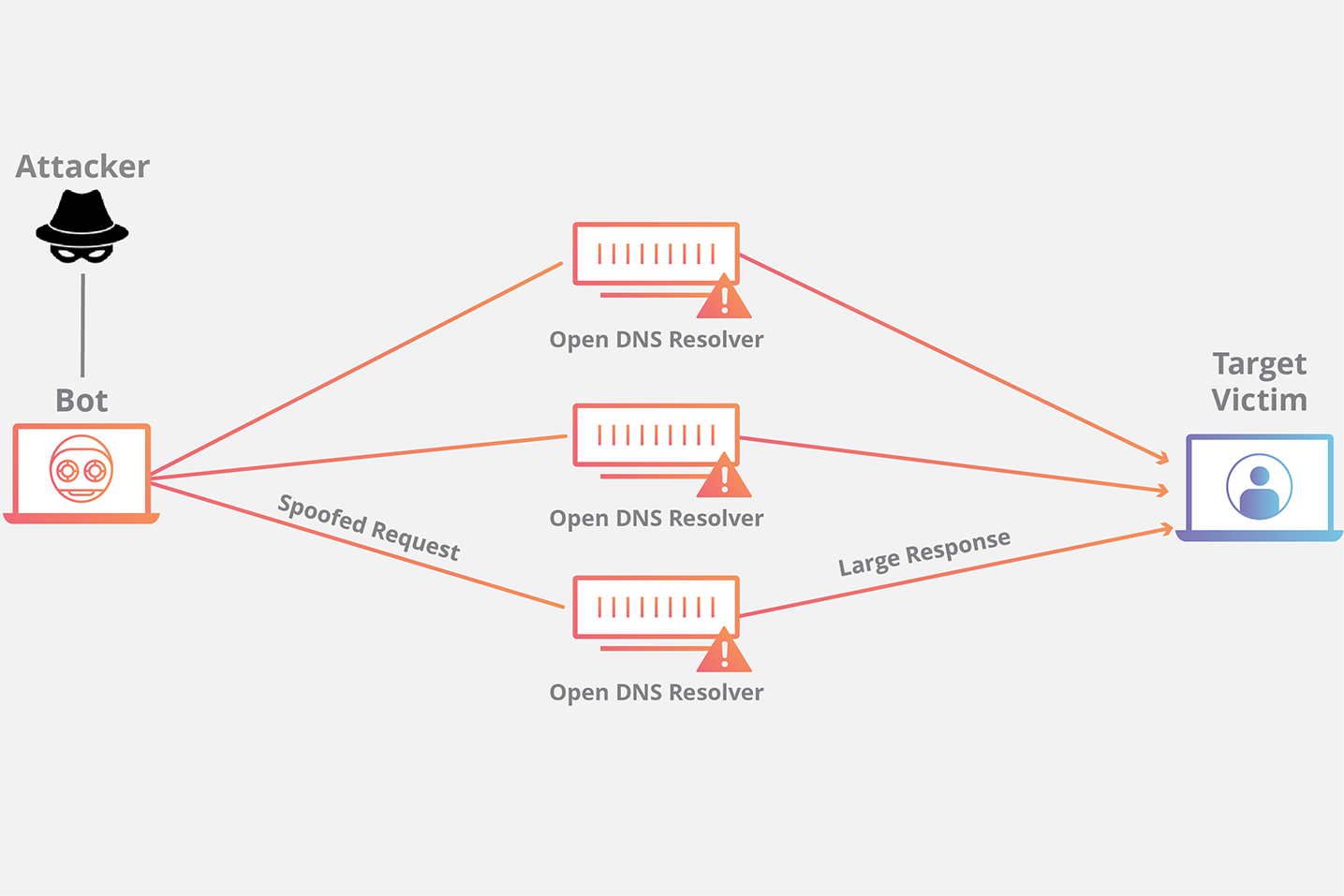
Hình 18: Kết quả tấn công TCP-SYN Flood bắt được trên Wireshark

* Hàng loạt gói tin TCP-SYN được gửi tới yêu cầu mở kết nối nhưng địa chỉ nguồn lại là địa chỉ giả mạo, khiến cho không tồn tại gói tin phản hồi TCP-ACK. Điều này vừa gây hao tốn tài nguyên bộ nhớ để chờ các gói tin phản hồi, vừa làm nghẽn băng thông mạng khiến cho các gói tin TCP-SYN hợp lệ khác không được đáp ứng.
* Lưu lượng tấn công thu được tại Wireshark lên tới hơn 17k pps:



Hình 19: Đo lưu lượng tấn công TCP-SYN Flood

* 1. DNS Amplification Attack
* DNS là máy chủ phân giải tên miền. Hệ thống này làm nhiệm vụ trả lời các yêu cầu về phân giải tên miền được gửi đến. Kịch bản của kiểu tấn công khuếch đại DNS này là kẻ tấn công sẽ giả mạo địa chỉ IP của nạn nhân, sau đó gửi hàng loạt yêu cầu phân giải tên miền tới một hoặc nhiều máy chủ DNS. Các máy chủ sẽ thực hiện giải mã tên miền và gửi kết quả về địa chỉ IP giả mạo trong gói tin, chính là địa chỉ IP của nạn nhân.
* Tấn công khuếch đại DNS là 1 dạng tấn công vào tầng ứng dụng, cũng vừa là một dạng tấn công vào băng thông mạng, khi mà nạn nhân nhận được hàng loạt kết quả phân giải tên miền từ máy chủ DNS, khiến cho các kết quả hợp lệ khác bị từ chối.
* Kịch bản tấn công
* Gửi hàng loạt gói tin yêu cầu phân giải tên miền, với giá trị IP Source là giá trị địa chỉ IP máy nạn nhân, tới DNS Server.
* DNS Server sẽ phân giải tên miền và gửi ngược lại nạn nhân.



Hình 20: Kịch bản tấn công DNS Amplification (Nguồn: CloudFlare)

* Công cụ thử nghiệm tấn công được sử dụng là dnsdrdos.c, tham khảo tại https://github.com/nullsecuritynet/tools/tree/master/dos/dnsdrdos
* Thử nghiệm:
* Máy chủ Window cài Wireshark.
* Máy tấn công Ubuntu.

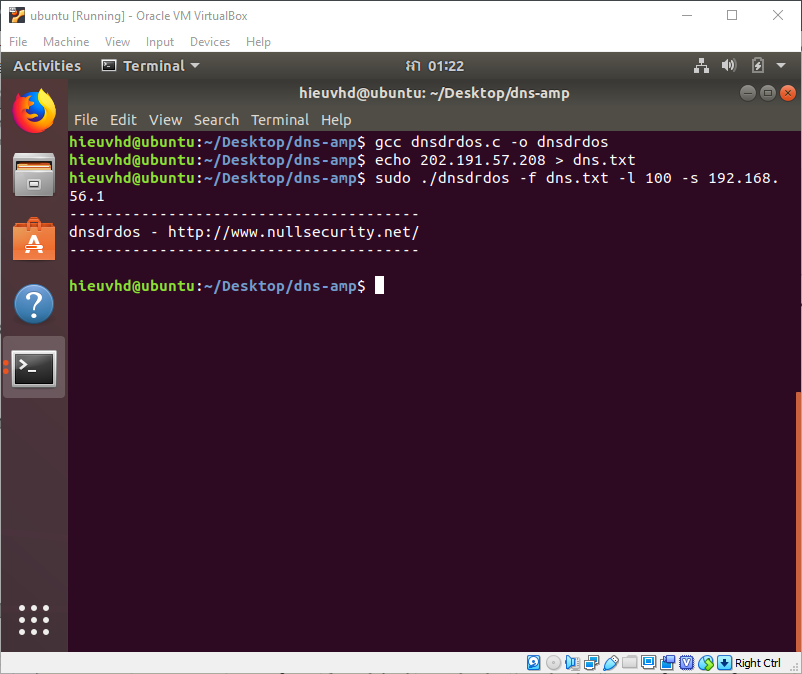
Tại máy tấn công, thực hiện tấn công:

Biên dịch chương trình: gcc dnsdrdos.c -o dnsdrdos

Tạo file text chứa địa chỉ IP của DNS Server: echo 202.191.57.208 > dns.txt

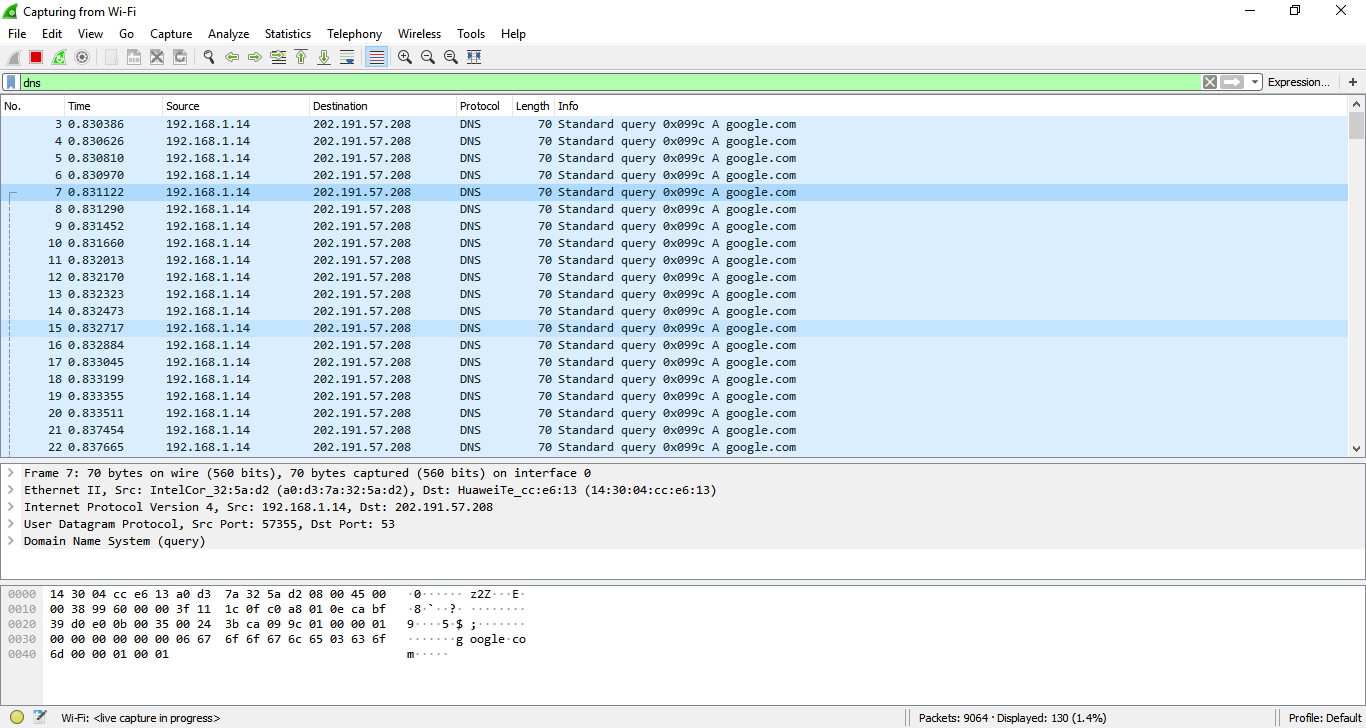
Thực hiện tấn công gửi 100 yêu cầu phân giải địa chỉ đến máy chủ 202.191.57.208 và gửi kết quả về địa chỉ giả mạo 192.168.56.1

sudo ./dnsdrdos -f dns.txt -l 100 -s 192.168.141.1



Hình 21: Tấn công DNS Amplification

* Kết quả thu được tại Wireshark khi bắt gói tin tại mạng WiFi:



Hình 22: Phân tích tấn công DNS Amplification

1. **TRIỂN KHAI MỘT SỐ BIỆN PHÁP PHÒNG THỦ**

Có nhiều phương pháp triển khai phòng thủ khác nhau tuỳ thuộc vào kiểu tấn công cũng như mục tiêu bị tấn công. Dưới đây là một số cách phòng chống tấn công DoS sử dụng công cụ netfilter/iptables, một công cụ để xây dựng các quy luật cũng như lọc gói tin của hệ điều hành Linux.

Về netfilter và iptables, đây là công cụ dung để thiết lập tường lửa của các hệ điều hành Linux, với khả năng lọc và phân tích các gói tin. Công cụ iptables giao tiếp với người dùng bằng các lệnh console cơ bản, từ đó thiết lập các luật gửi vào cho netfilter xử lí ngay tại nhân Linux. Netfilter làm việc trực tiếp với các gói tin IP trong nhân hệ điều hành nên tốc độ xử lí cực nhanh và gần như không ảnh hưởng tới tốc độ và tài nguyên hệ thống.

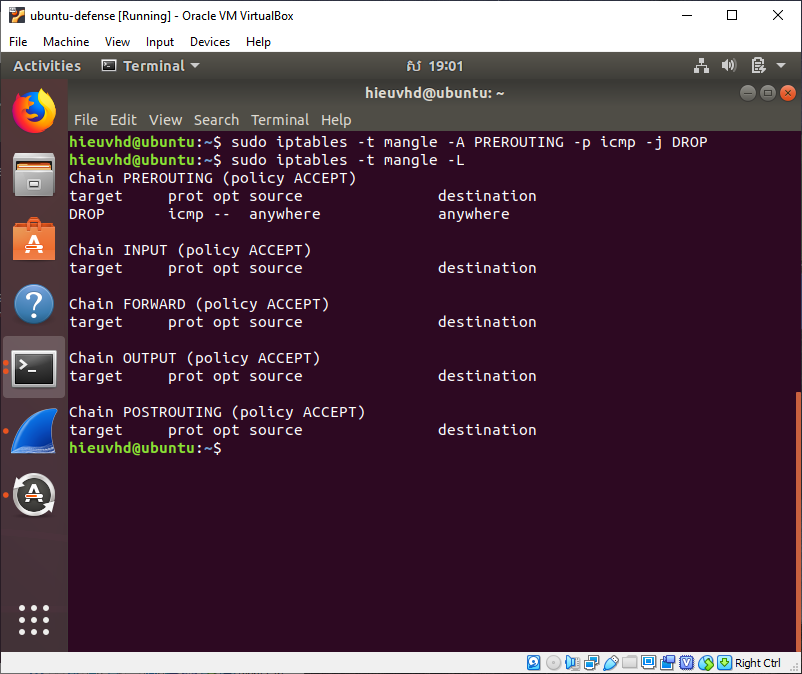
* 1. Ping of Death
* Để phòng thủ tấn công Ping of Death thì có nhiều cách. Cách đơn giản nhất là cập nhập hệ điều hành lên các phiên bản mới nhất, đã được vá lỗi khi gói tin ICMP gửi dữ liệu lớn hơn 65535 bytes. Đối với một số máy chủ, giao thức ICMP là không cần thiết thì chúng ta có thể loại bỏ tất cả các gói tin ICMP bằng table mangle của iptables.

Thêm luật vào iptables:

sudo iptables -t mangle -A PREROUTING -p icmp -j DROP

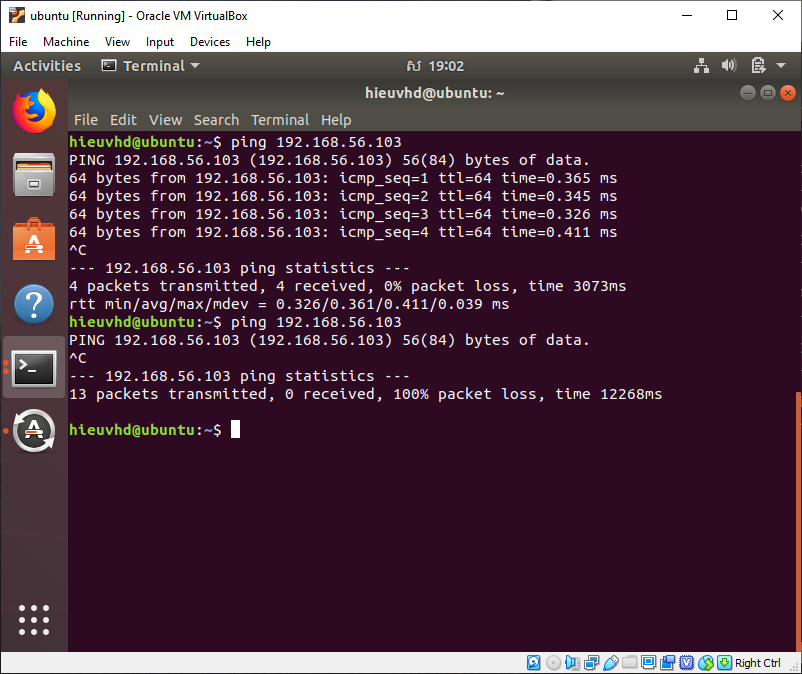
Hiển thị các luật của bảng mangle: sudo iptables -t mangle -L

Kết quả thu được:



Hình 23: Triển khai phòng chống Ping of Death

Thử nghiệm trên máy và thu được kết quả trước và sau khi thêm luật vào iptables.



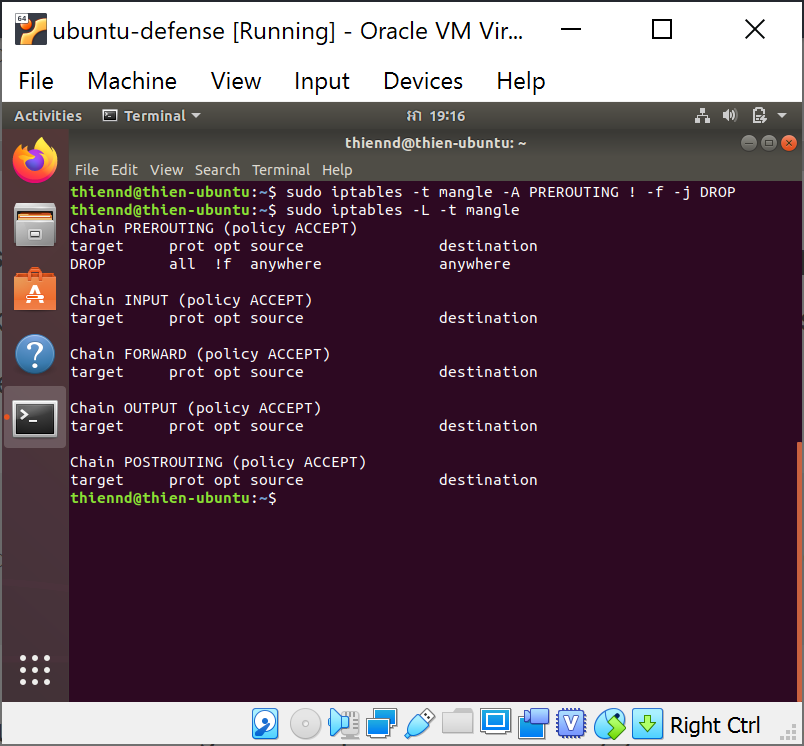
Hình 24: Thử nghiệm phòng chống Ping of Death

* Trong trường hợp vẫn muốn nhận gói tin ICMP, ta có thể thêm luật vào iptables để loại bỏ các gói tin TCP/IP mà có phần header MSS (Maximum Segment Size) không chính xác.
* Câu lệnh thêm lọc gói tin như sau:
* sudo iptables -t mangle -A PREROUTING -p tcp -m state --state NEW -m tcpmss ! --mss 536:65535 -j DROP
  1. Teardrop
* Để phòng chống tấn công Teardrop thì ta cần kiểm tra các giá trị offset nhận được trước khi thực hiện nối các mảnh dữ liệu lại.
* iptables cung cấp một lựa chọn giúp kiểm tra các gói tin phân mảnh, đó là --frag -f
* Câu lệnh thiết lập luật như sai:

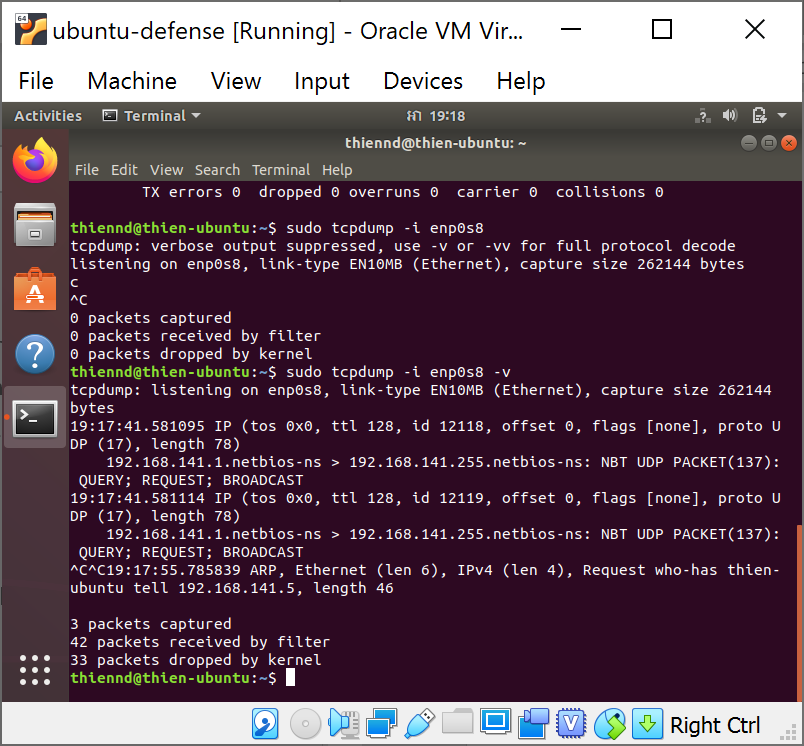
iptables -A PREROUTING ! -f -j DROP

* Lựa chọn -f ở đây sẽ áp dụng luật này với các gói tin bị phân mảnh và kiểm tra phần header xem offset có được đặt chính xác không, nếu không thì sẽ DROP tất cả các gói tin ngay tại kernel hệ thống.

Thử nghiệm: Sau khi thiết lập xong, ta thử tấn công lại và bắt kết quả trên máy phòng thủ bằng công cụ tcpdump để xem có bao nhiêu gói tin bị DROP.



Hình 25: Triển khai phòng chống Teardrop



Hình 26: Thử nghiệm phòng chống Teadrop

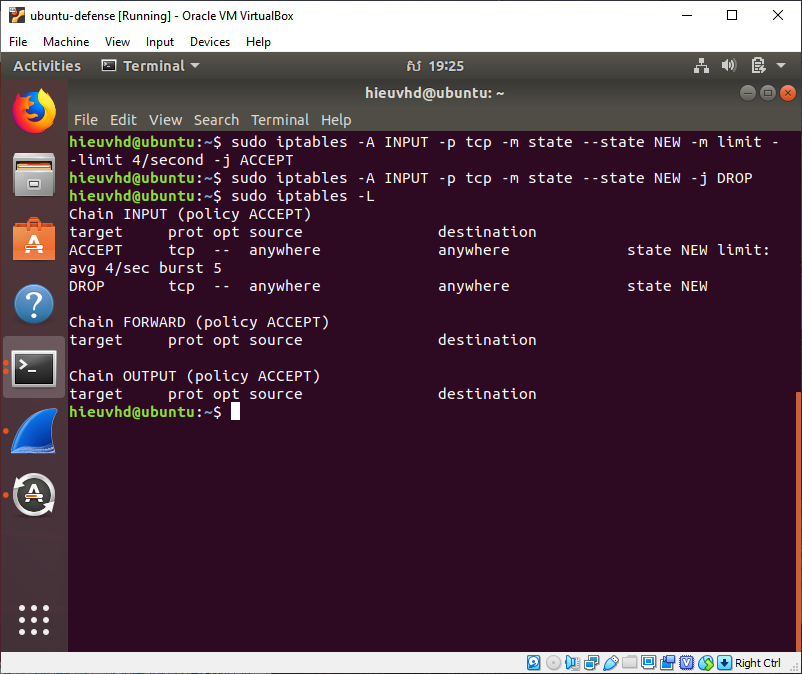
* Có tới 33 packets bị DROP bởi filter và kernel, đây là những packets bị sai phần offset trong header.
* Đây là một cách hiệu quả để phòng chống tấn công Teardrop nhưng điểm yếu của phương pháp này đó là chỉ phòng chống được gói tin phân mảnh lỗi của giao thức IPv4 chứ không kiểm soát được gói tin IPv6.
  1. TCP SYN Flood
* Với kiểu tấn công TCP SYN Flood thì rất khó có thể phòng chống hoàn toàn được, ta chỉ có thể hạn chế bằng cách giới hạn tốc độ cũng như số lượng kết nối được phép mở tại cùng một thời điểm.

sudo iptables -A INPUT -p tcp -m state --state NEW -m limit --limit 4/second --limit-rate 4 -j ACCEPT

* Câu lệnh trên sẽ giới hạn số lượng yêu cầu mở kết nối TCP xuống còn 4 kết nối mỗi giây.

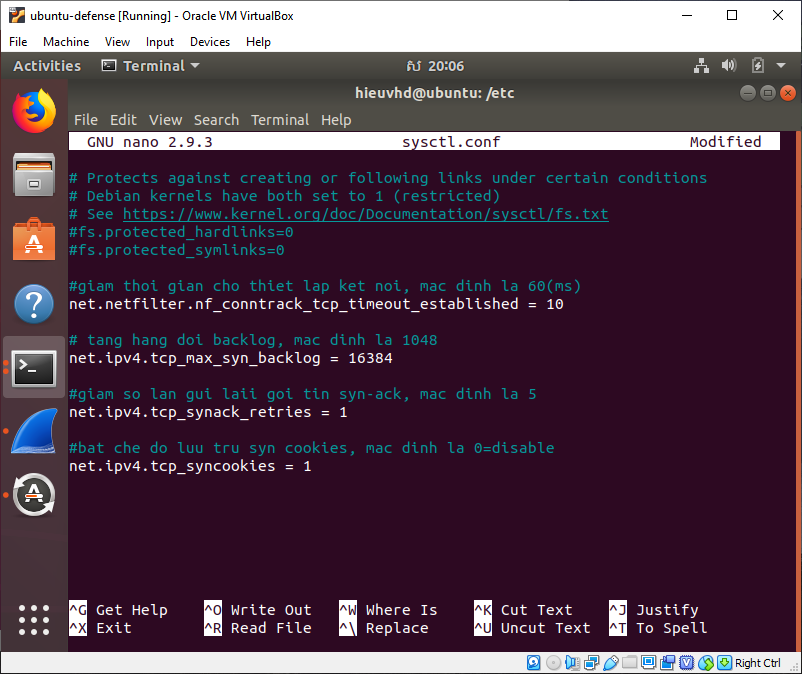
sudo iptables -A INPUT -p tcp -m state --state NEW -j DROP

* Sau đó các kết nối không thoả mãn sẽ bị drop. Lưu ý phải thực đúng thứ tự 2 lệnh.



Hình 27: Triển khai phòng chống TCP-SYN Flood

Ngoài ra, chúng ta còn có thể hạn chế tấn công TCP SYN Flood bằng cách tăng số lượng hàng đợi gói tin phản hồi TCP-ACK, giảm số lần gửi lại gói tin SYN-ACK, giảm thời gian chờ phản hồi thiết lập kết nối, sử dụng SYN Cookies lưu trữ cách kết nối đã được thiết lập. Các thiết lập này được thêm vào file /etc/sysctl.conf như sau:



Hình 28: Thử nghiệm phòng chống TCP-SYN Flood

Một số giải pháp khác nữa là server sẽ cố tình gửi gói tin SYN-ACK với tham số acknowledgement không đúng với gói tin TCP-SYN mà client đã gửi. Lúc này client hợp lệ sẽ gửi lại gói tin RST thông báo rằng không nhận được gói tin mong muốn và yêu cầu gửi lại. Lúc này server sẽ gửi gói tin đúng và thiết lập kết nối. Hiện nay một số thiết bị router cũng đã được trang bị sẵn các công cụ phòng chống các loại tấn công phổ biến như ICMP Flood, TCP Flood, …

* 1. DNS Amplification Attack
* Với kiểu tấn công khuếch đại này, ta có thể dùng iptables để từ chối tất cả các gói tin UDP hoặc TCP được gửi đến cổng 53 nếu máy chủ của chúng ta không cần sử dụng tới dịch vụ DNS.

sudo iptables -A INPUT -p udp --dport 53 -j DROP

sudo iptables -A INPUT -p tcp --dport 53 -j DROP

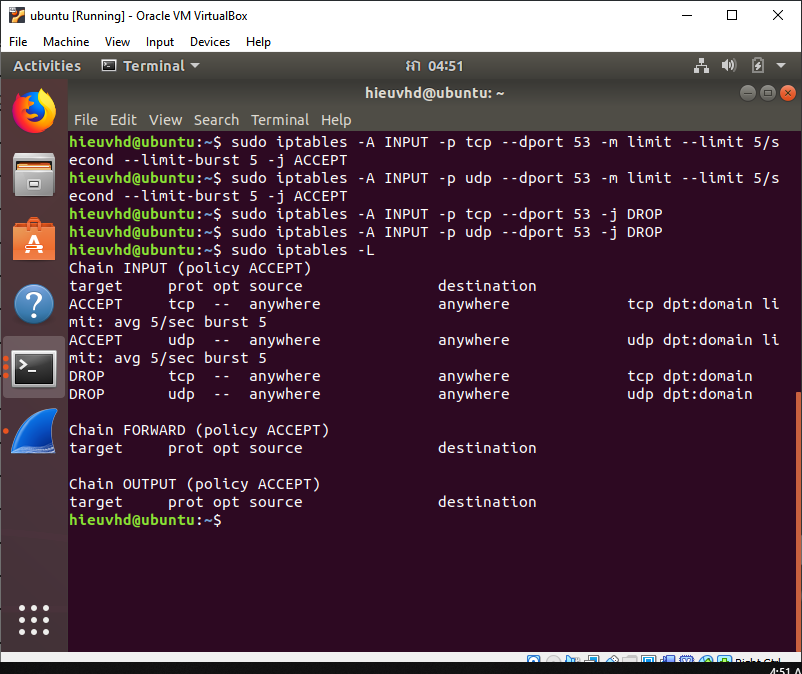
* Một giải pháp khác nếu như chúng ta vẫn cần sử dụng dịch vụ DNS tại cổng 53 này, mà vẫn muốn ngăn chặn tổn hại bởi tấn công khuếch đại DNS, đó là giới hạn số lượng gói tin tại cổng 53 lại. Nếu vượt quá giới hạn này thì các gói tin khác sẽ bị từ chối.

sudo iptables -A INPUT -p tcp --dport 53 -m limit --limit 5/second --limit-burst 5 -j ACCEPT

sudo iptables -A INPUT -p udp --dport 53 -m limit --limit 5/second --limit-burst 5 -j ACCEPT

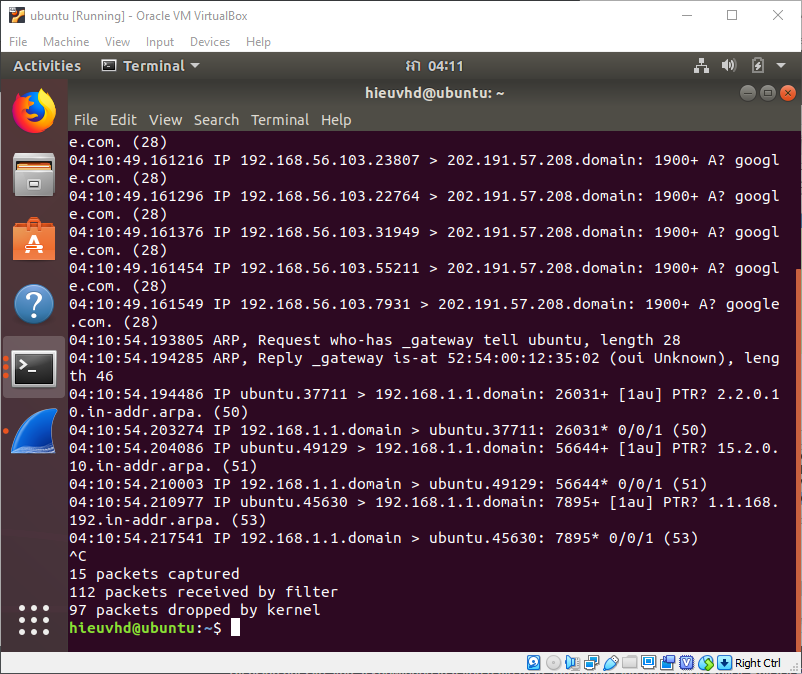
sudo iptables -A INPUT -p udp --dport 53 -j DROP

sudo iptables -A INPUT -p tcp --dport 53 -j DROP



Hình 29: Triển khai phòng chống DNS Amplification

* Kiểm tra lại kết quả sử dụng tcpdump:



Hình 30: Thử nghiệm phòng chống DNS Amplification

# **DANH MỤC TÀI LIỆU THAM KHẢO**

* + - 1. Anagnostopoulos, Marios, et al. "DNS amplification attack revisited." *Computers & Security* 39 (2013): 475-485.
      2. Hashmi, Mohd Jameel, Manish Saxena, and Rajesh Saini. "Classification of DDoS attacks and their defense techniques using intrusion prevention system." *International Journal of Computer Science & Communication Networks* 2.5 (2012): 607-614.
      3. Keith J. Gomes, J.D., Ph.D. (2016), Cybersecurity Cyber-Attack Series – Distributed Denial-Of-Service (DDoS),
      4. Elleithy, Khaled M., et al. "Denial of Service Attack Techniques: Analysis, Implementation and Comparison." (2005).
      5. Peng, Tao, Christopher Leckie, and Kotagiri Ramamohanarao. "Survey of network-based defense mechanisms countering the DoS and DDoS problems." *ACM Computing Surveys (CSUR)* 39.1 (2007): 3.
      6. Solankar, Prajakta, Subhash Pingale, and Ranjeetsingh Parihar. "Denial of Service Attack and Classification Techniques for Attack Detection." *(IJCSIT) International Journal of Computer Science and Information Technologies* 6.2 (2015): 1096-1099.
      7. Noction. “DDoS Amplification Attacks.” *Noction*, Noction Inc., 21 Aug. 2019, <https://www.noction.com/blog/ddos-amplification-attacks>, Accessed 20 Sep. 2019.
      8. “CYBERSECURITY CYBER-ATTACK SERIES DISTRIBUTED DEANIAL-OF-SERVICE.” *Xahive Inc*, Keith J. Gomes, J.D., Ph.D., 29 July 2016, <http://www.njamha.org/it/resources/XAHIVEFactsheetCyberAttacksDDOS.pdf>, Accessed 20 Sep 2019.
      9. “DOS vs DDOS Attacks: The Differences and How To Prevent Them.” *Comparitech*, Tim Keary, 30 July 2019, https://www.comparitech.com/net-admin/dos-vs-ddos-attacks-differences-prevention/.