BÁO CÁO BÀI TẬP

**Môn học: An toàn mạng máy tính**

**Kỳ báo cáo: Buổi 02 (Session 02)**

**Tên chủ đề: Lab2.2**

*GV: Nghi Hoàng Khoa*

*Ngày báo cáo: 20/10/2022*

**Nhóm: 9 (nếu không có xoá phần này)**

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT101.XXXX.YYYY

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Bình Thục Trâm | 20520815 | 20520815@gm.uit.edu.vn |
| 2 | Nguyễn Bùi Kim Ngân | 20520648 | 20520648@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Task 1 | 100% |
| 2 | Task 2 | 70% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

## Set up Environment

Bắt đầu ta tải file Labsetup.zip về, ta giải nén file này và sử dụng nó để build docker.

Ở đây, do docker version của em không tương thích, nên em đã chỉnh docker version trong bài lab từ “3” thành “3.7” và build thử thì căn bản là vẫn ổn nên em dùng luôn.

Build và compose-up xong thì em dùng ifcofig để kiểm tra mạng, đã thỏa yêu cầu của lab.

Text

Description automatically generated

## Task 1

## Tài nguyên: Ubuntu20.4 + Docker

## Mô tả/mục tiêu: Thực hiện được Sniffing & Spoofing bằng Scapy (Python)

## Các bước thực hiện/ Phương pháp thực hiện (Ảnh chụp màn hình, có giải thích)

*Task 1.1A: Run the program with the root privilege and demonstrate that you can indeed capture packets:*

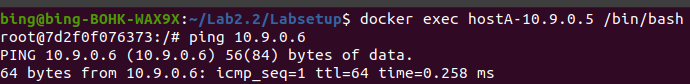
Sử dụng thư viện Scapy để viết một file python thực hiện yêu cầu đề bài.

Graphical user interface, text, website

Description automatically generated

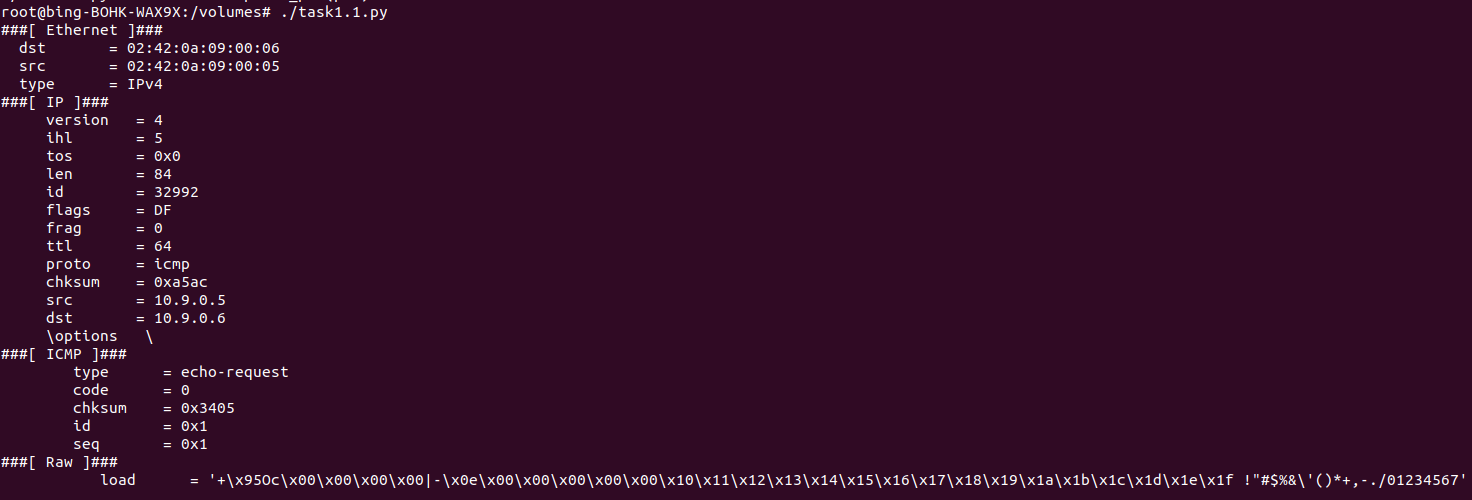
***Giải thích: Hàm sniff dùng để bắt các gói tin, iface là mạng, filter là để lọc các gói tin theo nhu cầu, prn là để gọi hàm print\_pkt() để in gói tin.***

Ta dùng lệnh docker exec để dùng hostA ping đến host B



Sau đó, ta sử dụng máy seed-attacker để chạy thử xem có bắt được các gói tin này hay không.

Các gói tin mà ta bắt được: Gói tin hostA ping đến hostB



Gói tin hostB reply cho hostA

A picture containing text

Description automatically generated

Chạy file khi không có quyền root:

Text

Description automatically generated

Khi chạy không có quyền root thì hệ thống đã báo lỗi, vì chỉ khi dùng root chúng ta mới xem được lưu lượng mạng qua lại trên máy. (Giống như khi sử dụng wireshark cũng cần dùng bằng lệnh sudo.)

*Task 1.1B: Please set the following filters and demonstrate your sniffer program again(each filter should be set separately):• Capture only the ICMP packet:*

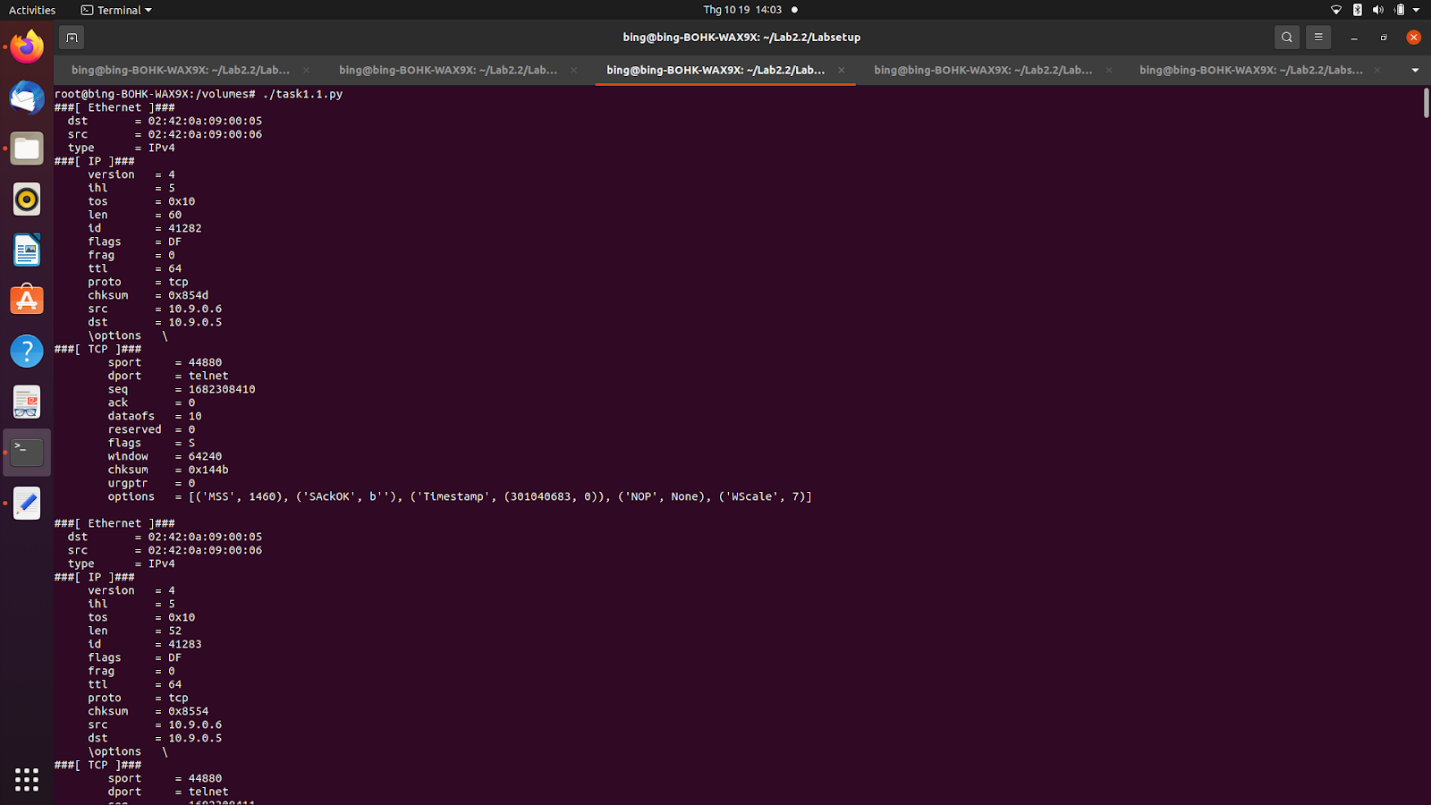
Ta làm tương tự với phần 1.1A và bắt được các gói tin ICMP từ 2 máy A B gửi cho nhau.

*• Capture any TCP packet that comes from a particular IP and with a destination port number 23.*

Ở đây, ta cũng code tương tự phần trên nhưng sẽ đổi filter lại. Theo yêu cầu của đề, chúng ta cần bắt các gói tin TCP từ một máy cố định gửi đến một máy khác trên port 23. Vì vậy, ta có thể suy ra filter như bên dưới:



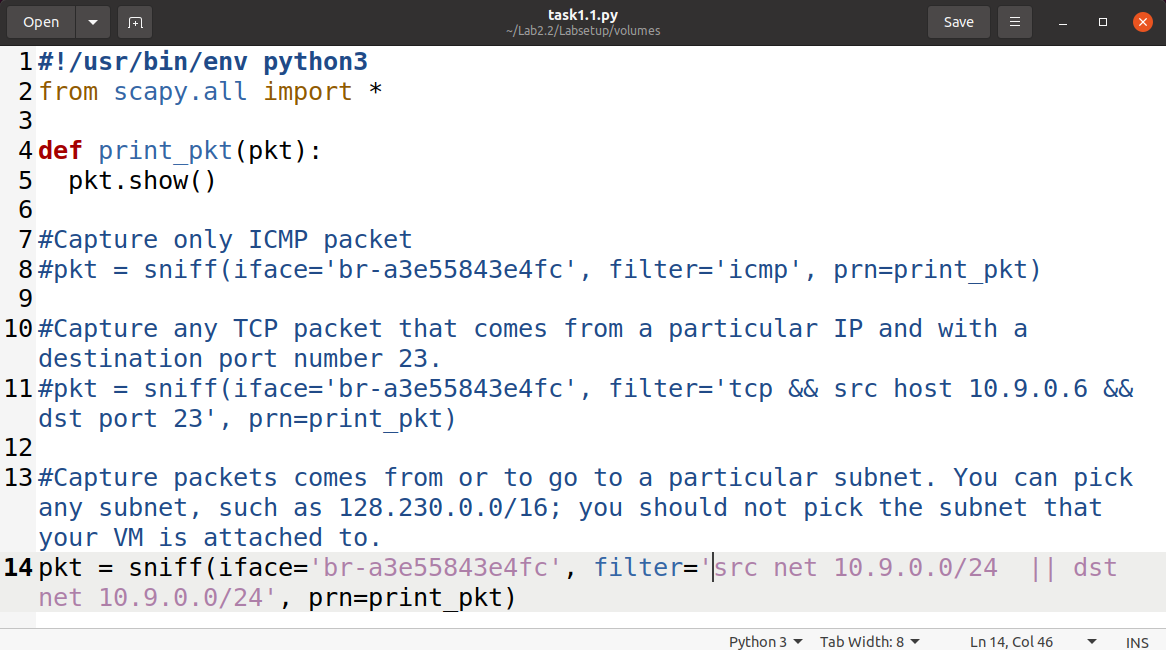
Để test thử xem có thể bắt được gói tin hay không, chúng ta cần cho hostB thử dùng Telnet để hostA và sau đó, chúng ta có thể xem gói tin bắt được trên attacker:



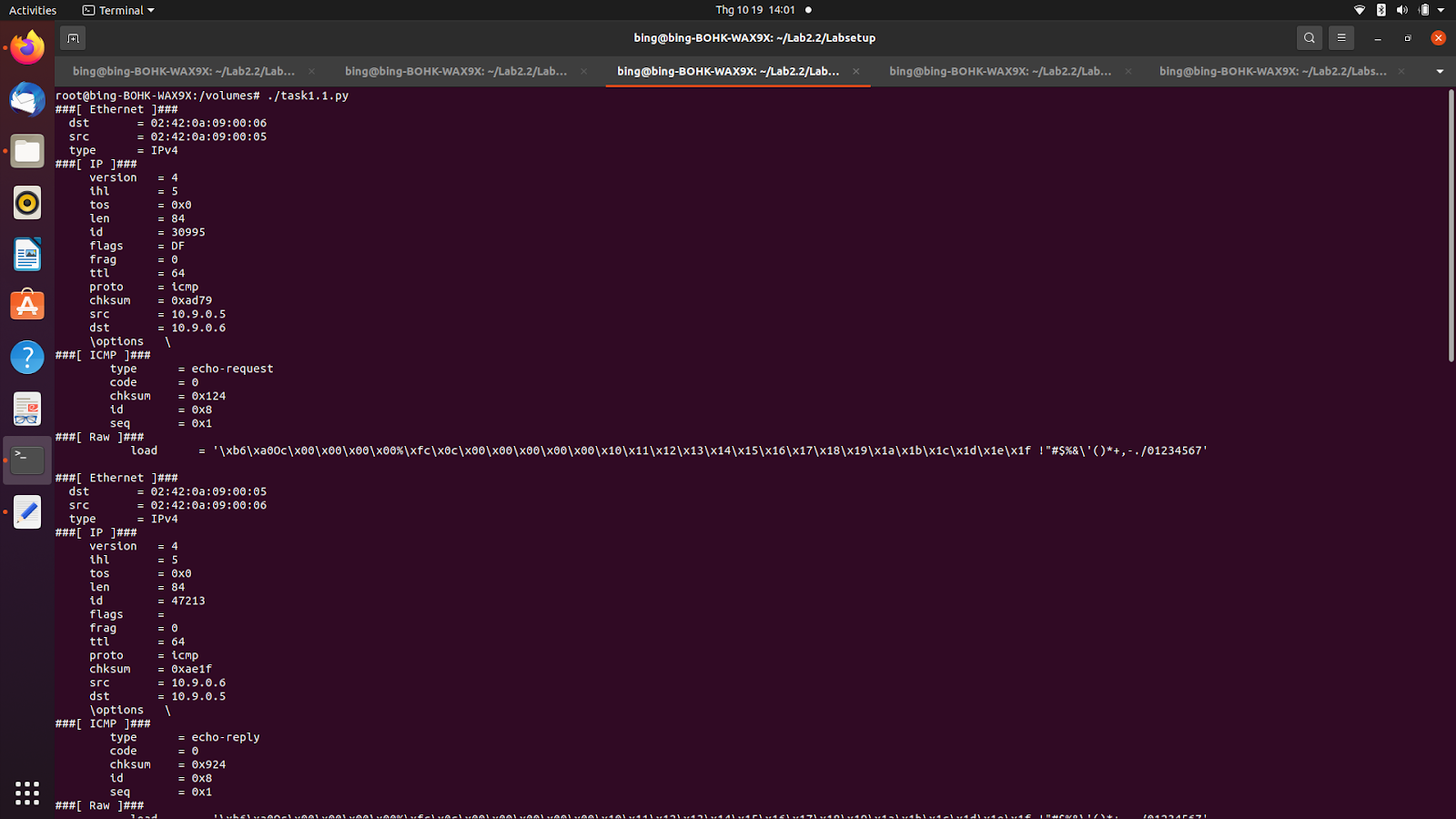
Ở đây, chúng ta có thể thấy được rằng, các gói tin bắt được đều dùng port telnet (23) và có source IP là 10.9.0.6, đúng với kì vọng.

*• Capture packets comes from or to go to a particular subnet. You can pick any subnet, such as 128.230.0.0/16; you should not pick the subnet that your VM is attached to.*

Phần kế tiếp này, chúng ta cũng làm tương tự với các phần trên, chỉ sửa lại filter cho đúng yêu cầu. Yêu cầu là muốn bắt các gói tin truyền trong cùng một mạng, vậy chúng ta có thể xác định được nội dung filter như sau:



Khi chạy thử, ta có thể thấy bên máy attacker đã bắt được những gói tin mà hostA và hostB gửi cho nhau:

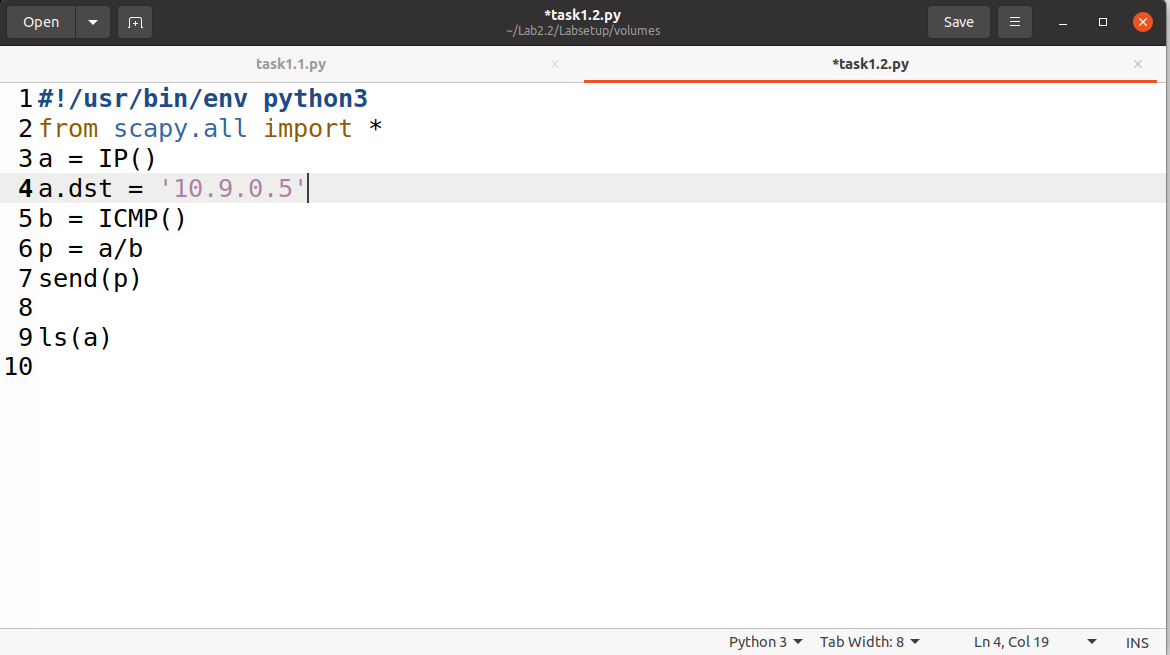


Text

Description automatically generated

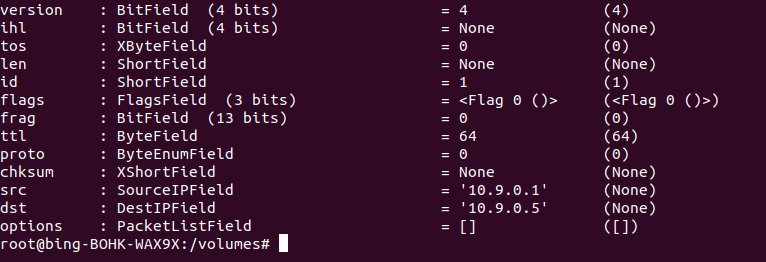
*Task 1.2: Spoofing ICMP Packets*

Ở phần này, ta bắt đầu thực hiện việc giả dạng gửi gói tin đến cho hostA.

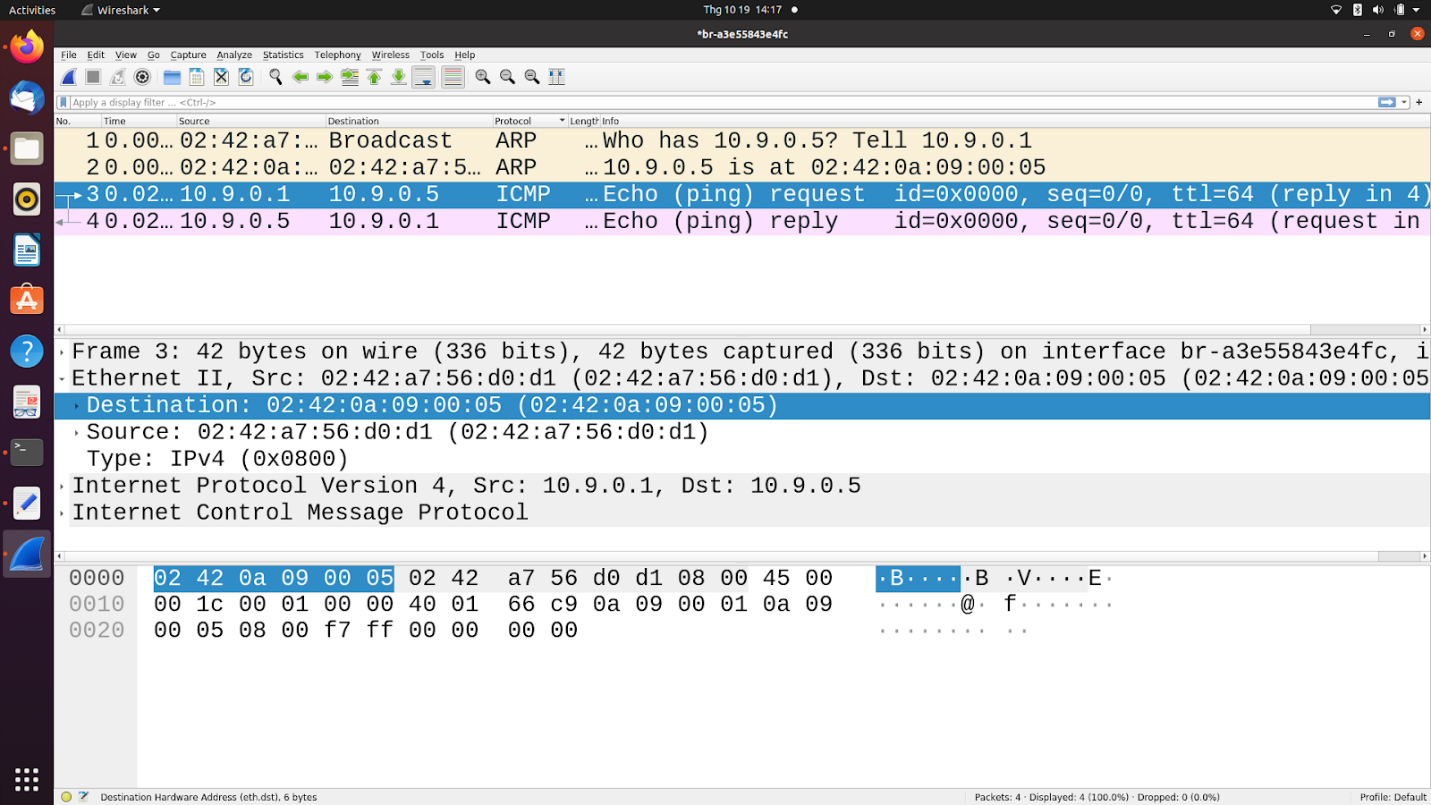


***Giải thích: Ở đây, chúng ta dùng a.dst để xác định IP máy nạn nhân, việc tiếp theo là sẽ gửi gói tin giả mạo ICMP cho nạn nhân bằng hàm send.***

***Hàm ls(a) cho phép ta kiểm tra các thông tin vừa được set up:***



Khi mở wireshark, chúng ta đã nhìn thấy gói tin ICMP mà ta vừa gửi:



*Task 1.3: Traceroute*

Ở đây yêu cầu chúng ta code một chương trình có thể xem được đường đi của các gói tin, tính khoảng cách từ máy ảo đến đích đến được chọn.

Graphical user interface, text, application

Description automatically generated

***Giải thích: Các gói tin được xem sẽ gửi từ máy của attacker đến trang “fb.com”. Lệnh sr1() sẽ gửi một gói tin và lắng nghe trên mạng 1 câu trả lời tương ứng, ở đây đặt thời gian timeout = 1 và cho verbose = 0 để nội dung trả về đơn giản nhất có thể.***

***Hàm sr1() sẽ bằng None khi không nhận được câu trả lời, vì vậy ta sẽ xác định là gói tin bị drop. Còn nếu nhận được câu trả lời thì ta sẽ print ra và tăng a.ttl lên (a.ttl là số bước nhảy qua router mà gói tin đã thực hiện được.)***

Text

Description automatically generated

Đây là khi chương trình chạy, ta có thể thấy được các gói tin chuyển thành công và các gói tin bị drop.

*Task 1.4: Sniffing and-then Spoofing*

Trong hình là các phân loại gói tin:

Graphical user interface, text, application, Word, email

Description automatically generated

Ở đây, vì đề bài yêu cầu thực hiện Sniffing và Spoofing đối với các gói tin Echo Reply và Echo Request nên ta cần chú ý để type 0 và type 8.

Ý tưởng ở đây khá đơn giản, chỉ là gộp 2 phần trước đó lại với nhau.

Ta có phần get\_pkt để bắt các gói tin, ở đây sẽ dùng filter là gói tin ICMP (Vì ping là ICMP), và sử dụng prn = spoofing là hàm để chúng ta gửi gói tin giả mạo.

Ở hàm spoofing, ta cần kiểm tra trước gói tin có đúng là gói tin Echo Request hay không bằng cách xem pkt[ICMP].type, khi đã đảm bảo đây là một gói tin để ping thì chúng ta bắt đầu giả mạo.

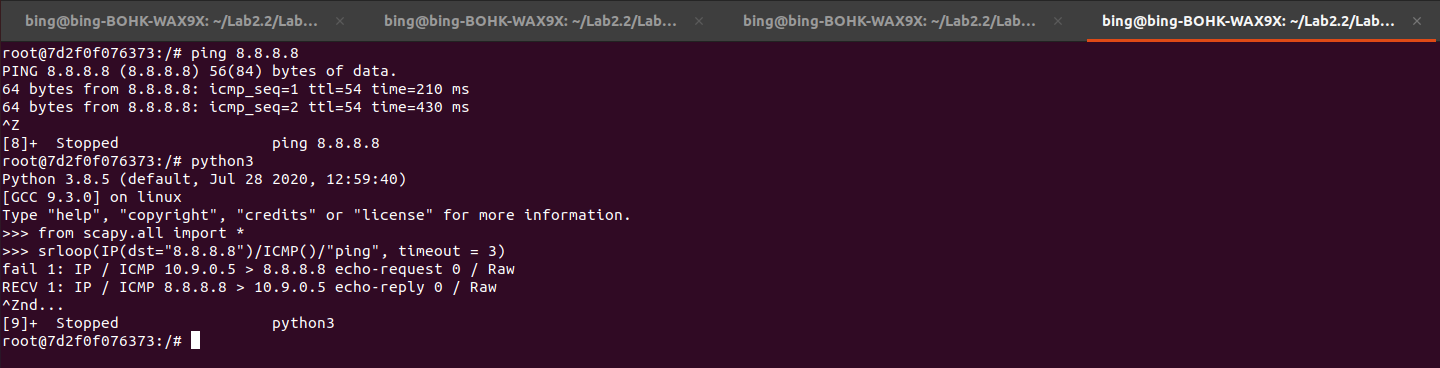
Trong phần setup của gói tin giả mạo, thay vì dùng IP của ta như trước thì ta sẽ dựa vào IP của gói tin đã bắt được, source IP sẽ là dest IP và ngược lại để giả vờ như một gói tin được gửi từ nơi mà nạn nhân ping đến. Đồng thời, các thông số khác như seq, id, ihl,… cũng phải được đảm bảo giống với gói tin đã bắt được.

Và quan trọng nhất, chúng ta phát gán type của gói tin này là ICMP type 0, để định dạng nó là một gói tin Echo Reply.

Graphical user interface, text

Description automatically generated

Bây giờ, chúng ta sẽ thử cho máy nạn nhân ping tới một địa chỉ bất kì.



Tiếp đó, chúng ta sẽ kiểm tra trong máy attacker:

Text

Description automatically generated

Có thể thấy rõ ràng, khi chạy chương trình, attacker đã bắt được gói tin của nạn nhân và gửi lại cho nạn nhân các gói tin echo reply.

## Task 2:

## Tài nguyên: Ubuntu20.4 + Docker

## Mô tả/mục tiêu: Thực hiện được Sniffing & Spoofing bằng pcap (C)

## Các bước thực hiện/ Phương pháp thực hiện (Ảnh chụp màn hình, có giải thích)

*Task 2.1: Writing Packet Sniffing Program*Set up device:

Graphical user interface, text, application

Description automatically generated

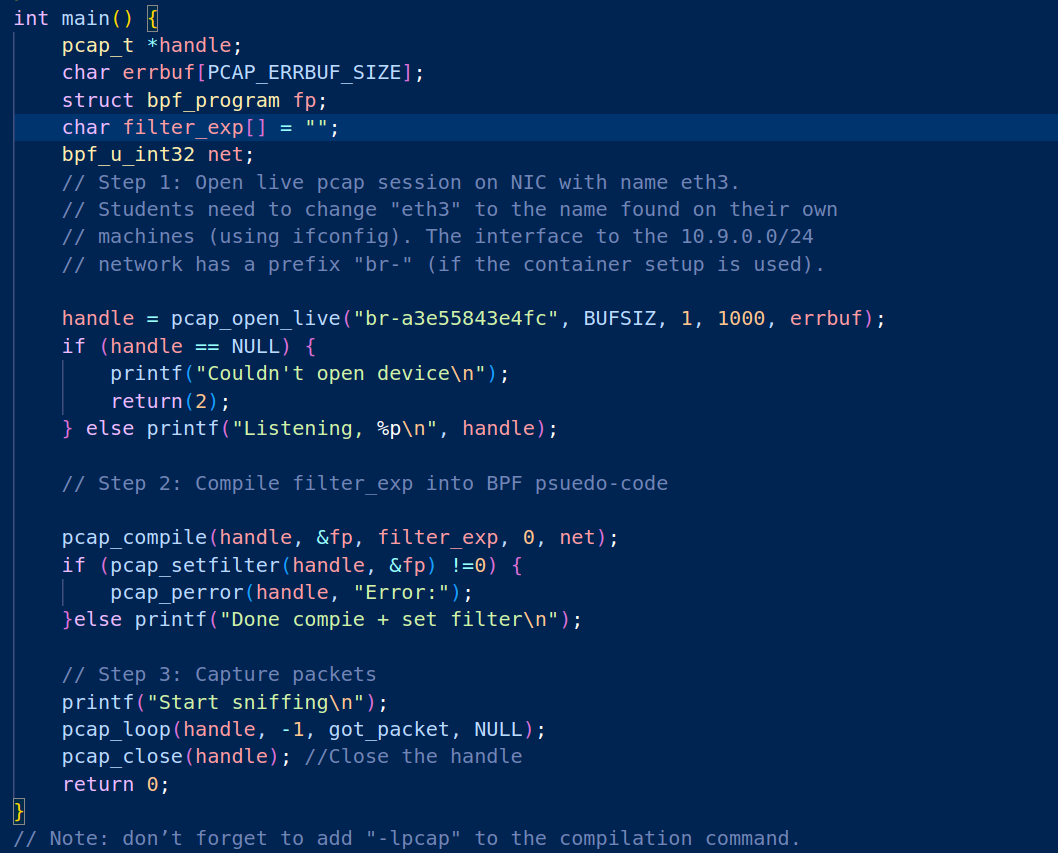
***Giải thích: Phần set up này giúp ta xác định được network card của các máy để hạn chế việc bị lỗi trong chương trình code kế tiếp.***

Tiếp theo, bài lab yêu cầu ta viết một chương trình bắt các gói tin bằng pcap. Sau đó tiếp tục nâng cấp, thêm filter vào trong chương trình.

Ở đây, chúng em làm gộp 2 chương trình này lại với nhau. Phần filter\_exp[] được bỏ trống để có thể tự tạo ra filter khác.

*Question 1. Please use your own words to describe the sequence of the library calls that are essential for sniffer programs.*

* Khởi động pcap trên NIC của máy bằng hàm pcap\_open\_live()
* Compile BPF filter bằng hàm pcap\_compile, sau đó dùng hàm set\_compile để set filter vào.
* Sử dụng hàm pcap\_loop để tạo ra một vòng lặp và bắt các gói tin.
* Sử dụng hàm pcap\_close để đóng kết nối, dừng việc bắt gói tin lại.



Tiếp theo, chúng ta có hàm got\_packet để thực hiện việc bắt các gói tin.

Graphical user interface, text

Description automatically generated

***Giải thích: Trong hàm này tích hợp việc bắt gói tin và filter các gói tin này theo các loại ICMP, TCP, UDP và một số loại khác. Khi bắt được mỗi gói tin, hàm sẽ in ra source IP và dest IP của nó, tiếp theo đó sẽ phân loại cho chúng ta.***

Bây giờ, chúng ta sẽ thử chạy chương trình. Đầu tiên, ta sẽ compile chương trình trên ubuntu trước, sau đó sẽ copy file exe vào docker để chạy trên máy của attacker.



Sau đó, chúng ta sẽ chạy file deviceSetup trước để kiểm tra lại NIC xem đã đúng chưa.



=> Nếu đã đúng thì chúng ta tiếp tục chạy chương trình trên máy attacker, đồng thời thực hiện ping từ hostA sang hostB để xem chương trình bắt được các gói tin.

Text

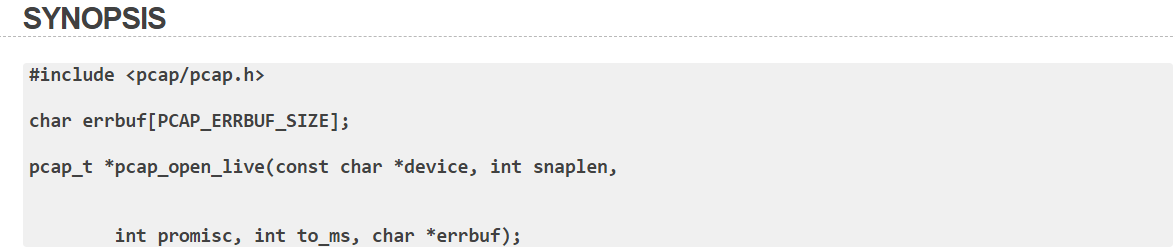
Description automatically generated

*Question 2. Why do you need the root privilege to run a sniffer program? Where does the program fail if it is executed without the root privilege?*

Việc bắt gói tin cần đến sử dụng quyền root, vì user mode không cho phép chúng ta có thể bắt được các gói tin do ở chế độ này sẽ không được phép truy cập vào NIC trong chế độ Promiscous. Vì vậy, chúng ta bắt buộc phải chạy chương trình sniffing với quyền root để chương trình được cấp đủ quyền thực thi, nếu không tới đoạn sniff gói tin thì sẽ gặp bug.

*Question 3. Please turn on and turn off the promiscuous mode in your sniffer program. The value 1 of the third parameter in pcap open live() turns on the promiscuous mode (use 0 to turn it off). Can you demonstrate the difference when this mode is on and off? Please describe how you can demonstrate this. You can use the following command to check whether an interface’s promiscuous mode is on or off (look at the promiscuity’s value).*

Để chỉnh được *promiscuous mode* trong chương trình, ta có thể tham khảm phần sau:



(Nguồn: <https://www.tcpdump.org/manpages/pcap_open_live.3pcap.html>)

Ở đây, chúng ta có thể thấy rằng tham số int promisc sẽ thay đổi promiscuous mode, 1 là bật và 0 là tắt. Khi bật chế độ này thì máy có thể bắt được tất cả các gói tin từ nguồn có cùng mạng với nó, nếu không thì chỉ có thể bắt các gói tin có liên kết trực tiếp với nó.

*Task 2.1B: Writing Filters. Please write filter expressions for your sniffer program to capture each of the followings. You can find online manuals for pcap filters. In your lab reports, you need to include screenshots to show the results after applying each of these filters.*

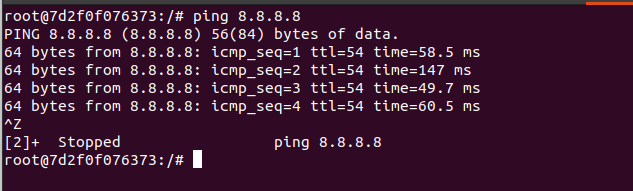
Đối với những yêu cầu bên dưới, chúng ta sử dụng phần code ở trên để làm, chỉ cần thay đổi biến filter\_exp[] để thay đổi điều kiện lọc là được.

*• Capture the ICMP packets between two specific hosts.*

Theo yêu cầu là ta sẽ bắt các gói ICMP giữa 2 host cố định, vậy ở đây là có thể chọn 1 host là Google 8.8.8.8 và hostA 10.9.0.5



Sử dụng máy để hostA thử:



Chương trình đã bắt được các gói tin đúng như trong kì vọng.

Text

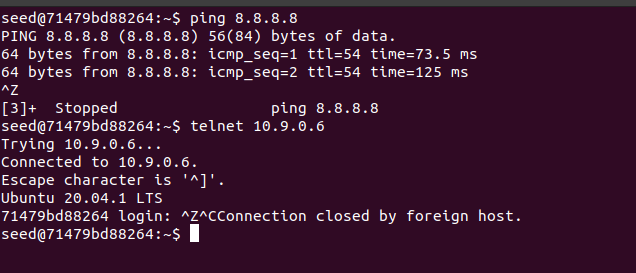
Description automatically generated

*• Capture the TCP packets with a destination port number in the range from 10 to 100.*

Ở đây, đề yêu cầu phải là TCP và port trong khoảng 10 – 100, ta có thể sử dụng portrange để filter



Ping + dùng Telnet trên hostA, nếu đúng thì chỉ hiển thị các gói tin Telnet:



Kết quả đúng như kì vọng:

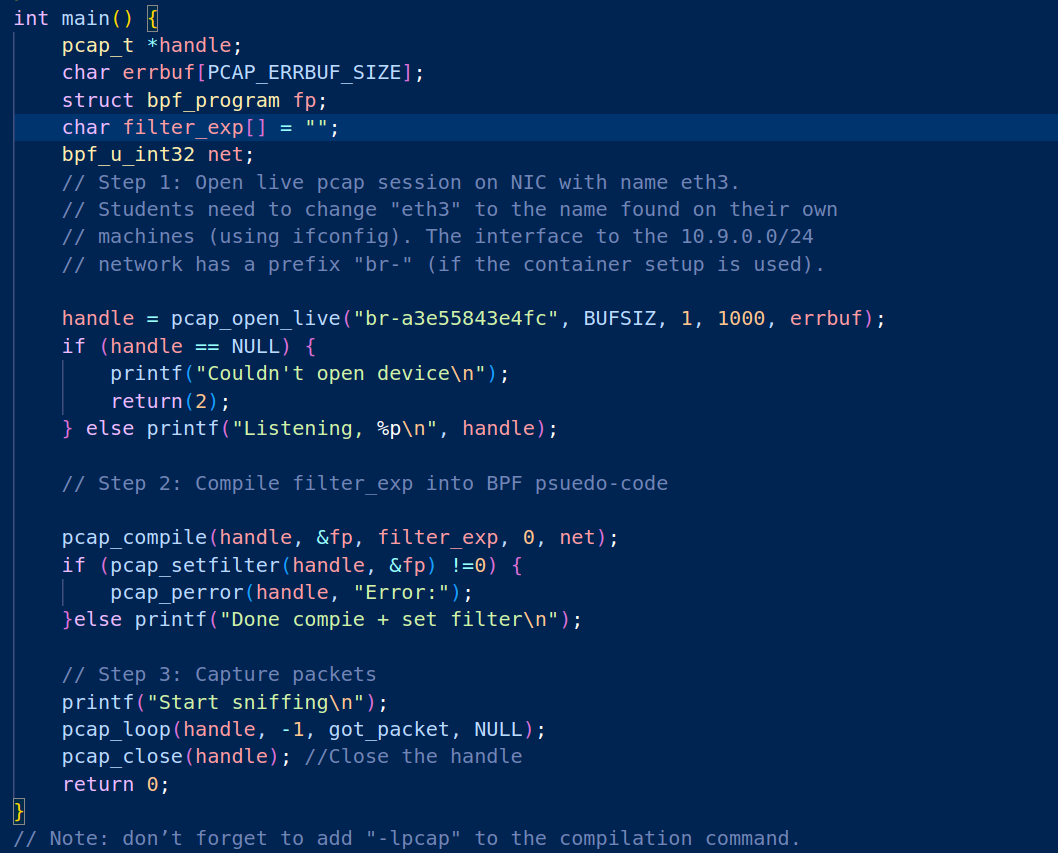
Shape, rectangle

Description automatically generated

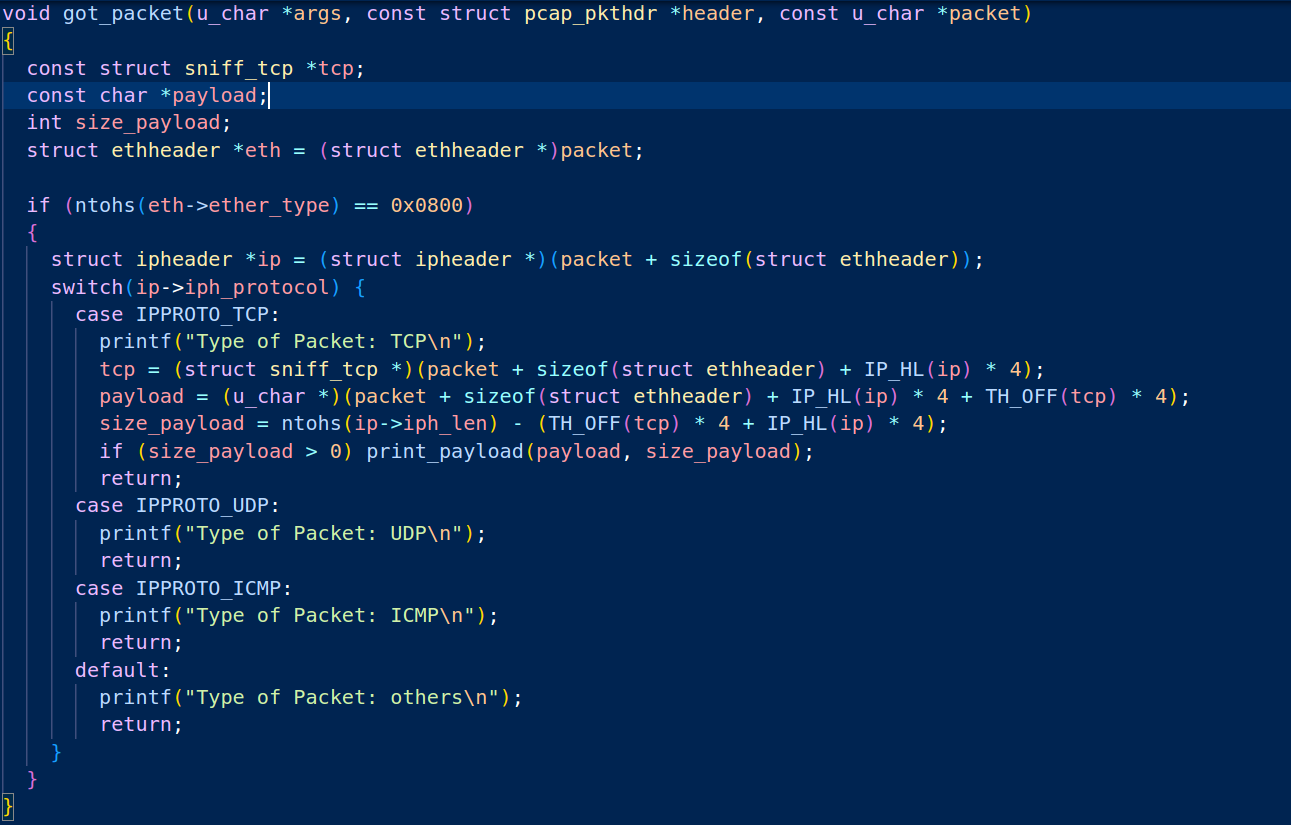
*Task 2.1C: Sniffing Passwords. Please show how you can use your sniffer program to capture the password when somebody is using telnet on the network that you are monitoring. You may need to modify your sniffer code to print out the data part of a captured TCP packet (telnet uses TCP). It is acceptable if you print out the entire data part, and then manually mark where the password (or part of it) is.*

Ở đây, đề yêu cầu chúng ta viết chương trình bắt gói tin TCP, đồng thời xem nội dung bên trong (password).

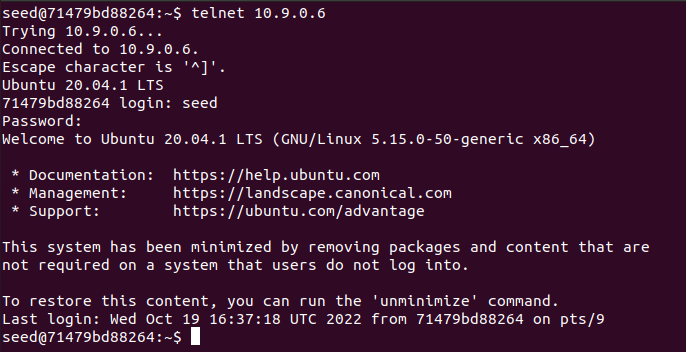
Vì là cùng bắt gói tin, nên ta có thể tận dụng hàm main bên trên, chỉ cần sửa lại một số thứ ở hàm got\_packet() để thực hiện được yêu cầu bắt gói TCP + xem được nội dung (payload).



Trong hàm got\_packet, chúng em đã thêm phần define cho gói tin TCP, từ đó define được cho payload của gói tin này. Payload sẽ ở dạng chuỗi, vì vậy chúng ta cũng có thể viết một hàm print chuỗi đơn giản để in nó ra.



HostA thực hiện telnet đến HostB:



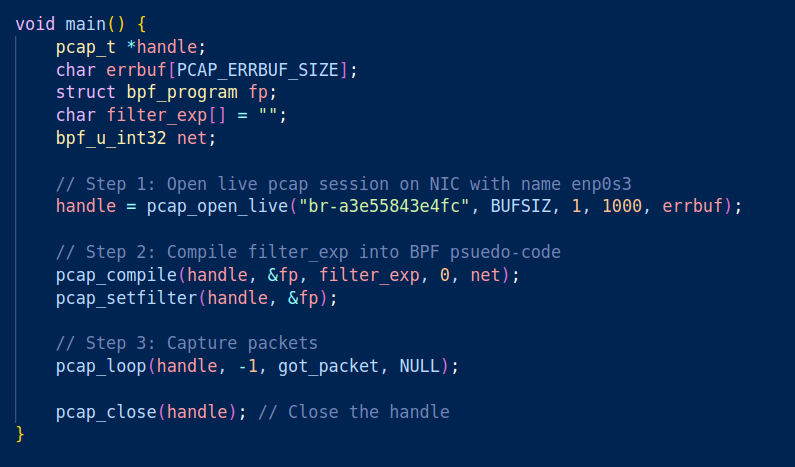
Attacker bắt được packet có kèm payload:

Text

Description automatically generated

*Task 2.2B: Spoof an ICMP Echo Request.*

Ở đây, hàm main tiếp tục là những bước quen thuộc để sniff một gói tin.



Sau khi sniff được gói tin ICMP để ping từ hostB, ta sẽ thực hiện việc spoof gói tin này để gửi lại cho hostA.

Ở đây, ta sử dụng thêm icmpheader để difine gói tin ICMP để có thể giả mạo.

Text

Description automatically generated

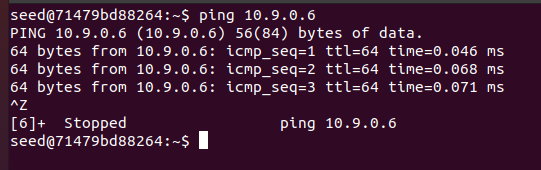
Sau khi set up xong mọi thứ, ta bắt đầu đưa ip vào hàm spoof để chương trình spoof gói tin theo các bước sau:

* Mở raw socket với IP Protocol
* Fill data structure để chuẩn bị gửi gói tin.
* Construct IP header, protocol header
* Gửi gói tin đi
* Đóng raw socket.

Text

Description automatically generated

Đầu tiên, ta cho hostA ping tới hostB



Sau đó, ta chạy chương trình bên attacker để thấy được chương trình đang spoofing các gói tin:

Text

Description automatically generated

*Question 4. Can you set the IP packet length field to an arbitrary value, regardless of how big the actual packet is?*

Có thể thay đổi chiều dài của gói tin. Nhưng chiều dài này sẽ bị sửa đè lên khi gói tin được gửi.

*Question 5. Using the raw socket programming, do you have to calculate the*

*checksum for the IP header?*

Khi dùng raw socket thì một số trường hợp kernal có thể tính toán checksum cho IP header.

*Why do you need the root privilege to run the programs that use raw sockets? Where does the program fail if executed without the root privilege?*

Cũng tương tự như ở trên, chỉ có quyền root mới cho phép ta chỉnh NIC ở chế độ promiscuous, đồng thời, chỉ có root mới có thể tùy ý chỉnh sửa các IP header nên nếu dùng user mode chạy chương trình thì sẽ bị lỗi ở chỗ sử dụng raw socket.

---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hành theo hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

**Báo cáo:**

* File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: [Mã lớp]-SessionX\_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).

*Ví dụ: [*NT101.K11.ANTT*]-Session1\_Group3.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**: Sinh viên hiểu và tự thực hiện. Khuyến khích:

* Chuẩn bị tốt.
* Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)