BÁO CÁO BÀI TẬP

**Môn học: An toàn mạng – NT140.O11.ANTT**

**Tên chủ đề: Packet Sniffing and Spoofing Lab**

*GV: Nghi Hoàng Khoa*

**Nhóm: 13**

1. **THÔNG TIN CHUNG:**

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

Lớp: NT140.O11.ANTT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Email |
| 01 | Đinh Bùi Huy Phương | 21520090 | 21520090@gm.uit.edu.vn |
| 02 | Nguyễn Thị Thanh Mai | 21521112 | 21521112@gm.uit.edu.vn |
| 03 | Lê Đoàn Trà My | 21521149 | 21521149@gm.uit.edu.vn |
| 04 | Nguyễn Phương Trinh | 21521581 | 21521581@gm.uit.edu.vn |

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

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nội dung |  | Trang |
| 01 | SET 1: USING SCAPY TO SNIFF AND SPOOF PACKETS | | 02-13 |
| 02 | SET 2: WRITING PROGRAMS TO SNIFF AND SPOOF PACKETS | | 14- |

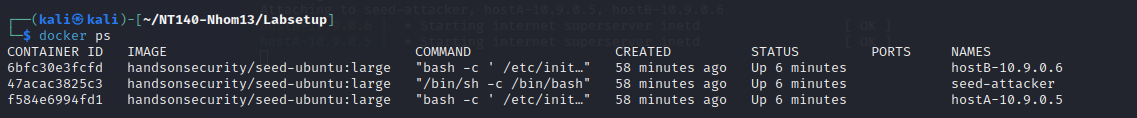
**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

\* Tiến hành Setup:

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

*Hình 1-4. Set up cơ bản*

**A. LAB TASK SET 1: USING SCAPY TO SNIFF AND SPOOF PACKETS**

**A screen shot of a computer screen

Description automatically generated**

*Hình 5. Sử dụng thử Scapy*

**1. Task 1.1: Sniffing Packets**

**A screenshot of a computer

Description automatically generated**

*Hình 6. Nội dung file Task1\_1.py*

a, Task 1.1.A: Trong chương trình trên, cho mỗi gói tin được bắt được, hàm gọi lại print\_pkt() sẽ được gọi; hàm này sẽ in ra một số thông tin về gói tin. Chạy chương trình với quyền root và chứng minh rằng thực sự có thể bắt được các gói tin. Sau đó, chạy chương trình một lần nữa, nhưng không sử dụng quyền root; mô tả và giải thích những quan sát của bạn.

***\* Thực thi với quyền root:***

- Trên terminal seed-attacker thực thi file Task1\_1.py và trên terminal hostA thực hiện ping đến hostB:

A screenshot of a computer

Description automatically generated

*Hình 7. Thực thi Task1\_1.py và trên seed-attacker và thực hiện ping từ hostA đến host B*

*A screen shot of a computer

Description automatically generated*

*Hình 8. Một phần của kết quả hiển thị trên terminal của seed-attacker với quyền root khi thực thi*

***\* Thực thi nhưng không sử dụng quyền root:***

***A computer screen shot of a code

Description automatically generated***

*Hình 9. Một phần của kết quả hiển thị trên terminal của seed-attacker khi thực thi không phải quyền root*

***\* Quan sát và giải thích:***

- Quan sát:

+ Khi chạy chương trình với quyền root, kết quả nhận được là thông tin về gói tin đã chỉ định trong hàm print\_pkt(), ví dụ: địa chỉ nguồn và đích, các trường dữ liệu trong gói tin, và các thông số khác liên quan.

+ Khi chạy chương trình với không phải quyền root, kết quả trả về là lỗi “Operation not permitted”, không có bất kì thông tin gói tin nào.

- Giải thích:

+ Khi chạy chương trình với quyền root, sẽ có quyền truy cập đầy đủ vào các giao diện mạng và có thể bắt được các gói tin.

+ Khi chạy chương trình mà không sử dụng quyền root, sẽ gặp các hạn chế trong việc truy cập vào các giao diện mạng. Scapy không có đủ quyền để mở một socket giao diện mạng RAW và bắt gói tin.

b, Task 1.1.B: Khi bắt gói tin, thường chỉ quan tâm đến một số loại gói tin cụ thể. Có thể làm điều đó bằng cách đặt bộ lọc trong quá trình bắt gói tin. Bộ lọc của Scapy sử dụng cú pháp BPF (Berkeley Packet Filter); có thể tìm thấy hướng dẫn BPF trên Internet. Vui lòng thiết lập các bộ lọc sau đây và thể hiện lại chương trình bắt gói tin (mỗi bộ lọc nên được thiết lập riêng biệt):

***\* Chỉ bắt gói tin ICMP:***

***A screen shot of a computer

Description automatically generated***

*Hình 10. Nội dung file Task1\_1.py để thực hiện bắt gói tin ICMP*

***A screenshot of a computer program

Description automatically generated***

*Hình 11. Một phần kết quả thực hiện bắt gói tin ICMP*

***\* Bắt bất kỳ gói tin TCP nào từ một địa chỉ IP cụ thể và với port đích là 23:***

- Để bắt gói tin từ hostB-10.9.0.6 và gói tin có port đích 23 thì trên hostB-10.9.0.6 cần đang chạy dịch vụ sử dụng port 23 (telnet) trên hostA-10.9.0.5.

A screenshot of a computer

Description automatically generated

*Hình 12. Nội dung file Task1\_1.py để thực hiện yêu cầu*

***A screenshot of a computer program

Description automatically generated***

*Hình 13. Kết quả thực hiện bắt gói tin TCP từ một IP cụ thể và với port đích là 23*

***\* Bắt gói tin từ hoặc đến một phân đoạn mạng cụ thể. Bạn có thể chọn bất kỳ phân đoạn mạng nào, ví dụ như 128.230.0.0/16; nhưng bạn không nên chọn phân đoạn mạng mà máy ảo của bạn đang kết nối đến:***

*A screenshot of a computer

Description automatically generated*

*Hình 14. Nội dung file Task1\_1.py để thực hiện yêu cầu*

*A screenshot of a computer

Description automatically generated*

*Hình 15. Kết quả thực hiện bắt gói tin trên phân đoạn mạng*

**2. Task 1.2: Spoofing ICMP Packets**

**A screenshot of a computer

Description automatically generated**

*Hình 16. Nội dung file Task1\_2.py*

***A screenshot of a computer

Description automatically generated***

*Hình 17. Gói tin Wireshark bắt được (từ eth0 và filter = host 10.0.2.3) và kết quả khi thực thi Task1\_2.py*

**3. Task 1.3: Traceroute**

**A screenshot of a computer

Description automatically generated**

*Hình 18. Nội dung file Task1\_3.py*

**A screen shot of a computer

Description automatically generated**

*Hình 19. Tiến hành thực thi file Task1\_3.py với ttl được nhập lần lượt từ bàn phím*

- Tại ttl = 2, nhận được packet từ địa chỉ 31.13.77.35 (facebook), tức là đã tiếp cận 2 bộ định tuyến.

**4. Task 1.4: Sniffing and-then Spoofing**

**a, Ping đến 1.2.3.4**

**A screenshot of a computer screen

Description automatically generated**

*Hình 20. Nội dung file Task1\_4.py để ping đến host 1.2.3.4*

**A screenshot of a computer

Description automatically generated**

*Hình 21. Kết quả thu được khi thực thi Task1\_4.py và các gói tin Wireshark bắt được khi thực hiện ping 1.2.3.4 trên hostB*

→ Khi ping 1.2.3.4 (một máy chủ không tồn tại trên Internet), gói tin được gửi đi sẽ mất hoàn toàn và không có phản hồi nào về nguồn, tuy nhiên chương trình sniff-and-then-spoof gửi đi một gói tin ICMP echo reply giả mạo. Chương trình ping trong container người dùng sẽ nhận được phản hồi, cho thấy 1.2.3.4 đang hoạt động. Tuy nhiên, thực tế 1.2.3.4 không tồn tại, vì vậy phản hồi là giả mạo.

**b, Ping đến 10.9.0.99**

*A computer screen shot of a program code

Description automatically generated Hình 22. Nội dung file Task1\_4.py để ping đến host 1.2.3.4*

**A screenshot of a computer

Description automatically generated**

*Hình 23. Kết quả thu được khi thực thi Task1\_4.py và các gói tin Wireshark bắt được khi thực hiện ping 10.9.0.99 trên hostB*

→ Khi ping 10.9.0.99 (một máy chủ không tồn tại trên mạng LAN), theo lí thuyết thì trường hợp 2 giống với khi ping 1.2.3.4. Tuy nhiên trong Wireshark, nhóm em chỉ thấy các gói tin ARP và không hề thấy bất cứ một gói tin ICMP nào được gửi đi và phản hồi (có thể là gói ICMP giả mạo hoặc không) của nó.

**c, Ping đến 8.8.8.8**

**A screenshot of a computer program

Description automatically generated**

*Hình 24. Nội dung file Task1\_4.py để ping đến host 8.8.8.8*

**A screenshot of a computer

Description automatically generated**

*Hình 25. Kết quả thu được khi thực thi Task1\_4. và các gói tin Wireshark bắt được khi thực hiện ping 8.8.8.8 trên hostB*

→ Khi ping 8.8.8.8 (một máy chủ tồn tại trên Internet), máy chủ sẽ nhận được gói tin ICMP echo request và gửi gói tin ICMP echp reply lại đồng thời chương trình sniff-and-then-spoof cũng gửi đi một gói tin ICMP echo reply giả mạo. Chương trình ping trong container người dùng sẽ nhận được phản hồi trùng lặp như trên.

**B. LAB TASK SET 2: WRITING PROGRAMS TO SNIFF AND SPOOF PACKETS**

**1. Task2.1: Writing Packet Sniffing Program**

a, Task 2.1.A: Understanding How a Sniffier Works

***Question 1.***

***Question 2.***

***Question 3.***

b, Task 2.1.B: Writing Filters

c, Task 2.1.C: Sniffing Passwords

**2. Task 2.2: Spoofing**

a, Task 2.2.A: Write a spoofing program

b, Task 2.2.B: Spoof an ICMP Echo Request

***Question 4.***

***Question 5.***

***Question 6.***

**3. Task 2.3: Sniff and then Spoof**

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