Bài thực hành: Giấu tin trong trường định danh IPv4

1. Khái quát chung

1.1. Tổng quan

Bài lab này giúp tìm hiểu một kỹ thuật giấu tin trong mạng dựa trên trường Identification (ID) của gói tin IPv4. Đây là một phương pháp giấu tin trong header giao thức mà không làm thay đổi nội dung của gói tin, giúp che giấu dữ liệu một cách khó phát hiện hơn.

1.2. Nguyên lý giấu tin

- Trường IP ID (Identification) trong header IPv4 có độ dài 16 bit, thường được sử dụng để nhận diện và tái lắp ráp các gói tin bị phân mảnh.
- Trong kỹ thuật giấu tin này, thay vì sử dụng IP ID một cách ngẫu nhiên hoặc tuần tự như thông thường, kẻ tấn công sẽ mã hóa thông điệp cần giấu vào giá trị IP ID của gói tin trước khi gửi đi.
- Cụ thể, mỗi ký tự trong thông điệp được chuyển thành giá trị số ASCII, sau đó nhóm thành từng cặp (mỗi cặp 2 ký tự sẽ tạo thành một số 16-bit) và gán vào IP ID của gói tin.

1.3. Mục tiêu của bài lab

- Hiểu được cách giấu tin trong header gói tin IPv4 bằng cách kiểm soát trường Identification
- Hiểu cách hoạt động của Scapy để tạo và gửi gói tin chứa thông điệp bí mật
- Sử dụng Wireshark để phân tích lưu lượng mạng, nhận diện và tách thông tin giấu tin

1.4. Yêu cầu đối với sinh viên

- Sinh viên đã có kiến thức về giấu tin trong mạng, hiểu về cấu trúc gói tin IP và trường ID của tiêu đề IP.
- Có kiến thức cơ bản về hệ điều hành Linux, công cụ Wireshark.

2. Chuẩn bị môi trường thực hành

Mở terminal, trong thư mục labtainer-student, bắt đầu bài thực hành bằng lệnh:

labtainer -r stego-code-identification_ipv4

*(chú ý: sinh viên sử dụng **mã sinh viên** của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)*

Kiểm tra các bài tập phải làm trong bài thực hành

Checkwork

3. Nội dung thực hành

3.1. Mã hóa nhị phân thông điệp bí mật

Chuyển đổi thông điệp thành nhị phân và lưu vào file binary.txt. Sinh viên đã được cung cấp một script Python có tên encrypt.py. Trên terminal của máy sender gõ lệnh *ls* để kiểm tra.

Bên máy sender, chạy lệnh sau để tiến hành mã hóa nhị phân thông điệp:

python3 encrypt.py

Chương trình này sẽ chuyển từng ký tự trong thông điệp thành dạng nhị phân (8-bit mỗi ký tự) rồi lưu vào binary.txt.

3.2. Chuyển thành IP ID, chèn nhiễu, gửi gói tin.

Sinh viên sẽ phải thực hiện việc chuyển nhị phân trong file binary.txt vừa mới thực hiện bên thành IP ID, rồi thực hiện nhiễu và gửi gói tin đi. Lưu ý rằng sinh viên đã được cung cấp một script Python có tên sender.py.

sender.py sẽ đọc binary.txt, chuyển nhị phân thành IP ID (mỗi 2 ký tự thành 16-bit), rồi thực hiện nhiễu bằng cách tăng thêm số lượng packet không chứa thông điệp rồi thực hiện random vị trí của các packet chứ thông điệp (vẫn đảm bảo thứ tự trước sau của thông điệp cũng như packet để bên nhận còn có thể ghép lại được) và gửi gói tin đi.

Bên máy receiver, sử dụng lệnh sau để khởi chạy wireshark:

wireshark &

Trên máy sender, chạy lệnh

python3 sender.py

- Kết quả mong đợi: Quan sát wireshark bên máy receiver, xác nhận gửi gói tin thành công.

3.3. Giải mã thông điệp bí mật được giấu

Sau khi giấu tin vào IP ID và gửi đi thành công, bên nhận phải tiến hành giải mã thông điệp bí mật.

Bên máy receiver, chạy lệnh:

python3 receiver.py

Quan sát trong wireshark, tìm ra các packet nào là packet có giấu tin bên trong. Sau đó, nhập các giá trị IP ID, cách nhau bằng dấu phẩy (*chú ý: nhập theo đúng thứ tự các packet gửi đi để giải mã và ghép lại đúng thông điệp bí mật*), rồi nhấn Enter để giải mã thông điệp.

Gợi ý: cách lọc gói tin chứa dữ liệu giấu

1. Trong Wireshark, nhập bộ lọc sau vào thanh **Filter** (lọc gói tin):

   icmp && ip.src == <IP_Máy sender>

   (Trong bài thực hành này IP_Máy sender là : **172.20.0.2**)

   → Bộ lọc này giúp hiển thị **các gói ICMP** do chương trình gửi đi.

2. Kiểm tra từng gói tin, tìm trường IP Identification (ID).

- Nhấn vào một gói tin ICMP
- Mở phần **Internet Protocol Version 4 (IPv4)**
- Xem giá trị **Identification** (IP ID)

- Kết quả mong đợi: Tìm ra được hide message.

4. Kết thúc bài lab

- **Kết thúc bài lab:**
    - Kiểm tra checkwork:

      checkwork

    - Trên terminal đầu tiên sử dụng câu lênh sau để kết thúc bài lab:

      stoplab

    - Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.
- **Khởi động lại bài lab:**
    - Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

      labtainer -r stego-code-identification_ipv4