Bài thực hành: Tấn công Giả mạo Phá hoại Kênh Giấu tin trong trường định danh IPv4

1. Khái quát chung

1.1. Tổng quan

Sinh viên sẽ đóng vai kẻ tấn công trong một mạng mô phỏng, nơi hai máy (Sender và Receiver) trao đổi lưu lượng IPv4 chứa thông điệp ẩn trong trường Identification. Nhiệm vụ là thực hiện tấn công MITM để:

- Chặn các gói tin IPv4 hợp pháp giữa Sender và Receiver, sau đó tiến hành giải mã thông điệp ban đầu từ Sender ("*SECRET*").
- Giả mạo gói tin để thay thông điệp "*SECRET*" (Identification) thành "*HACKED*".
- Đảm bảo Receiver nhận được thông điệp sai lệch thay vì thông điệp gốc.

Sinh viên sẽ sử dụng **Scapy** để thực hiện ARP spoofing và gửi gói tin giả mạo, cùng với **Wireshark** để quan sát lưu lượng và kiểm tra kết quả. Bài lab kết thúc bằng việc thông điệp sai lệch mà Receiver nhận được.

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
- Phát triển kỹ năng thực hiện tấn công man-in-the-middle (MITM) để giả mạo gói tin IPv4.
- Học cách phá hoại kênh giấu tin bằng cách thay đổi thông điệp ẩn trong trường Identification.
- 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, ARP spoofing.

2. Chuẩn bị môi trường thực hành

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

labtainer -r stego-attk-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. Chặn và giải mã thông điệp ban đầu

Trên terminal của máy **attacker** sinh vien phải thực hiện thiết lập MITM, chặn lưu lượng từ Sender, và giải mã thông điệp ban đầu ("*SECRET*").

Trong bài lab này sẽ có 2 terminal attacker, sinh viên có thể dùng bất kỳ terminal nào cũng được. Để thiết lập 1 cuộc tấn công man-in-the-middle (MITM) thì trên terminal attacker thứ nhất, sinh viên thực hiện lệnh:

sudo python3 arpspoof.py --mode spoof -t <IP_Sender> -g <IP_Receiver>

(Trong bài thực hành này **IP_Sender** là : **172.20.0.2**, **IP_Receiver** là: **172.20.0.3**)

Sinh viên phải giữ nguyên lệnh này cho tới khi kết thúc cuộc tấn công MITM. Lệnh này dùng để chạy script arpspoof.py ở chế độ spoof, trong đó attacker sẽ liên tục gửi các gói ARP giả mạo để đánh lừa cả **sender** và **receiver**, khiến chúng gửi dữ liệu qua máy attacker. Đồng thời sẽ chặn hoàn toàn luồng giao tiếp giữa hai máy này bằng cách thêm rule iptables và tắt IP forwarding, tạo điều kiện để attacker thực hiện chặn, đọc hoặc chỉnh sửa thông điệp truyền đi.

Trên terminal attacker thứ hai, sinh viên tiến hành thực hiện lệnh:

sudo python3 arpspoof.py --mode decode

Lệnh này dùng để chạy script arpspoof.py ở chế độ giải mã (decode), trong đó attacker sẽ bật chế độ sniff gói tin ICMP trên mạng, trích xuất và giải mã thông điệp ẩn được giấu trong trường IP ID của các gói tin đó. Thông điệp sẽ được hiển thị ra màn hình khi người dùng nhấn **Ctrl+C** để dừng.

Sau khi chế độ giải mã(decode) thực hiện thì nó sẽ chờ đợi để bắt gói tin chứa thông điệp mà sender thực hiện gửi qua cho receiver.

3.2. 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`.

3.3. Chuyển thành IP ID, 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) 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 các gói tin đã bị chặn do cuộc tấn công ARP Spoof.

Ngay sau khi thực hiện gửi gói tin đi thì quay trở lại terminal attaker thứ 2, sinh viên sẽ thấy attacker đã chặn và giải mã gói tin thành công. Nhấn **Ctrl+C** để dừng sniffing, thông điệp bí mật sẽ được hiển thị ra màn hình.

3.4. Thực Hiện Tấn Công MITM để Thay Đổi Thông Điệp

Bây giờ, sinh viên sẽ thực hiện việc thay đổi thông điệp từ "SECRET" thành "HACKED" bằng cách gửi các gói tin giả mạo.

Trên terminal attacker thứ 2, sinh viên chạy lệnh:

sudo python3 arpspoof.py --mode modify -t <IP_sender> -g <IP_receiver> --message HACKED

(Trong bài thực hành này **IP_Sender** là : **172.20.0.2**, **IP_Receiver** là: **172.20.0.3**)

Lệnh này dùng để chạy script arpspoof.py ở chế độ modify, trong đó sẽ gửi một thông điệp đã chỉnh sửa ("*HACKED*") từ attacker đến máy receiver (IP: 172.20.0.3), giả mạo rằng nó đến từ sender (IP: 172.20.0.2). Thông điệp được mã hóa vào trường IP ID của các gói ICMP và gửi đi từng cặp ký tự một cách tuần tự.

3.5. Giải mã thông điệp bên phía Receiver

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.dst == <IP_Receiverr>

(Trong bài thực hành này IP_Receiver là : **172.20.0.3**)

→ Bộ lọc này giúp hiển thị **các gói ICMP**.

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 (thông điệp này đã bị sửa đổi do attacker can thiệp).

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-attk-identification_ipv4