HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN

BÁO CÁO BÀI THỰC HÀNH
HỌC PHẦN: MẬT MÃ HỌC CƠ SỞ
MÃ HỌC PHẦN: INT1344

XÂY DỰNG BÀI THỰC HÀNH TRÊN NỀN TẢNG LABTAINER
TẤN CÔNG MD5 COLLISION TRÊN CHỮ KÝ SỐ

Sinh viên thực hiện:

B22DCAT076 – Nguyễn Hữu Đạt

B22DCAT132 – Phí Công Huân

Giảng viên hướng dẫn: PGS.TS Đỗ Xuân Chợ

HỌC KỲ 2 NĂM HỌC 2024-2025

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH

1.1 Mục đích

Bài thực hành nhằm giúp sinh viên:

- Hiểu rõ cơ chế hoạt động của chữ ký số RSA và vai trò của hàm băm trong việc bảo đảm tính toàn vẹn dữ liệu.
- Nhận biết lỗ hổng nghiêm trọng khi sử dụng các hàm băm yếu như MD5 trong hệ thống xác thực chữ ký số.
- Mô phỏng một cuộc tấn công thực tế sử dụng MD5 collision, trong đó kẻ tấn công thay thế một tập tin hợp lệ bằng một tập tin độc hại có cùng mã băm mà vẫn vượt qua được quá trình xác thực.
- Nâng cao kỹ năng thao tác với môi trường ảo Labtainer, sử dụng các lệnh mạng, tạo và xác thực chữ ký số, sử dụng SSH và công cụ scp.

1.2 Tìm hiểu lý thuyết

- Chữ ký số RSA là một cơ chế mã hóa sử dụng khóa bất đối xứng để xác nhận tính toàn vẹn và nguồn gốc dữ liệu. Quy trình gồm:
    - Tạo một bản băm của dữ liệu (bằng hàm băm).
    - Mã hóa bản băm đó bằng khóa riêng để tạo chữ ký số.
    - Người nhận sử dụng khóa công khai để giải mã chữ ký, rồi so sánh với giá trị băm tính lại từ dữ liệu gốc.
- Hàm băm (Hash function) là hàm một chiều, dùng để tạo ra "dấu vân tay" duy nhất cho dữ liệu. Một hàm băm lý tưởng phải:
    - Khó đoán ngược (pre-image resistance).
    - Không thể tìm được hai dữ liệu khác nhau cùng cho ra một mã băm (collision resistance).
- MD5 là một hàm băm phổ biến trước đây nhưng hiện đã bị phá vỡ tính toàn vẹn. Các nhà nghiên cứu đã chứng minh có thể tạo hai file khác nhau có cùng MD5 hash (collision), từ đó có thể tấn công các hệ thống xác thực chữ ký nếu vẫn dùng MD5.
- Trong bài thực hành, sinh viên sẽ khai thác điểm yếu của MD5 để đánh tráo một file độc hại vào vị trí file ban đầu, nhưng hệ thống vẫn xác thực chữ ký là hợp lệ – qua đó thấy được tính chất nguy hiểm của việc dùng thuật toán băm yếu trong an ninh mạng.

CHƯƠNG 2. NỘI DUNG THỰC HÀNH

2.1 Các bước thực hiện

- Khởi động bài lab

  imodule https://github.com/chupinana04/Labtainer/raw/refs/heads/main/lab3.tar

- Sinh viên khởi động bài lab

  labtainer -r md5-col-atk-digital-sign

  - (Chú ý: sinh viên sử dụng MÃ SINH VIÊN của mình để nhập thông tin người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm.)
  - Sau khi khởi động xong 3 terminal ảo sẽ xuất hiện "bob", "alice", "attacker".
  - Kịch bản bài lab: Bob đang muốn gửi file ảnh chứa nội dung quan trọng cho Alice, Bob sử dụng chữ ký số RSA để đảm bảo tính toàn vẹn khi gửi file ảnh đó, trong quá trình tạo chữ ký số sử dụng hàm băm MD5 (một hàm băm yếu). Attacker tạo ra một file ảnh có nội dung độc hại có cùng giá trị băm với file ảnh quan trọng trên, và cố gắng tráo đổi nó. Do cùng có giá trị băm nên file ảnh độc hại dễ dàng có thể vượt qua được hệ thống xác nhận chữ ký số trên máy Alice.

2.1.1 TASK 1

- Sinh viên sử dụng lệnh *ifconfig* để kiểm tra ip của 3 terminal và sử dụng lệnh ls để xem mỗi terminal có những gì
- Trên terminal "bob" sinh viên sử dụng ***feh*** để xem file message.png cần gửi:

feh message.png

- Sinh viên cần kiểm tra trên terminal Bob ở thư mục hiện tại đã có file tao_khoa_rsa.py để tạo cặp khóa RSA
- Sinh viên chạy file tao_khoa_rsa.py để tạo cặp khóa riêng và công khai bằng lệnh:

python3 tao_khoa_rsa.py

- Sinh viên nhập độ dài khóa rsa (bội của 1024).
- Tạo thành công khóa riêng và khóa công khai, sử dụng ls để xem.
- Sinh viên tạo một bản chữ ký số với ảnh message.png và khóa riêng tư vừa tạo, sinh viên mở file với lệnh:

nano tao_chu_ky_so.py

- Sinh viên chỉnh sửa đường dẫn của các file code sao cho phù hợp
- Sinh viên chạy file tao_chu_ky_so.py để kí chữ kí số vào file message.png bằng lệnh:

python3 tao_chu_ky_so.py

- Sau khi tạo chữ ký số thành công, sinh viên cần gửi file gốc, file chữ ký số và khóa công khai cho Alice
- Sinh viên cần kiểm tra dịch vụ ssh đã active hay chưa

sudo systemctl status ssh

- Sinh viên sử dụng scp để copy file message.png, message.sig và khóa công khai sang "alice"

scp "file gốc" "chữ ký số" "khóa công khai" ubuntu@<ip_alice>:~/

- Trên terminal "alice" sử dụng lệnh ***ls*** để kiểm tra các file đã được chuyển thành công.

2.1.2 TASK 2

- Trên terminal "attacker" gồm message-good.png và message-toxic.png để thực hiện tráo đổi file.
- Sinh viên dùng lệnh **feh** để xem nội dung hai ảnh
- Sinh viên sử dụng lệnh md5sum để kiểm tra giá trị băm của 2 ảnh có trùng khớp nhau không?

md5sum <tên file>

- Sinh viên thay đổi tên file toxic cho giống tên file message trên terminal "alice" để thực hiện đánh tráo
- Sinh viên sử dụng lệnh scp để tráo file đã đổi tên sang terminal "alice":

scp "message" ubuntu@<ip_alice>:~/

2.1.3 TASK 3

- Alice xác thực file message nhận từ Bob (file gốc đã bị attacker đánh tráo)
- Sinh viên chạy file xac_thuc.py để xác thực.
- Hệ thống thông báo chữ ký hợp lệ, mặc dù file message ban đầu đã bị đánh tráo.
- Alice mở file ảnh message.png sau khi xác thực
- Tại sao file message.png đã bị đánh tráo nhưng hệ thống vẫn xác thực chữ ký hợp lệ? Tìm hiểu cách hoạt động của chữ kỹ số RSA sử dụng hàm băm MD5 ?

2.1.4 Kết thúc lab:

- Trên terminal khởi động lab, sinh viên sử dụng lệnh:

stoplab

- Khi bài lab kết thúc, một tệp lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab. Sinh viên cần nộp file .lab để chấm điểm.
- Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh:

checkwork md5-col-atk-digital-sign

- Khởi động lại bài lab: Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

labtainer -r md5-col-atk-digital-sign