PROG3: SSH LOGGER

# Lý thuyết

## Tìm hiểu cách hoạt động của dịch vụ sshd và tiến trình ssh

* **sshd** ( OpenSSH Deamon): là tiến trình trên máy chủ, lắng nghe kết nối SSH (thường qua cổng 22), xác thực người dùng, thiết lập phiên làm việc và xử lý lệnh từ xa.
* **ssh** (SSH Client): là chương trình trên máy khách, giúp kết nối đến máy chủ SSH. Hoạt động của **ssh** gồm : gửi yêu cầu kết nối → xác thực → thiết lập phiên bảo mật → truyền lệnh/dữ liệu → kết thúc.

## Tìm hiểu cơ chế xác thực của PAM của hệ điều hành linux

* **PAM**( Pluggable Authentication Modules): PAM là hệ thống mô-đun xác thực linh hoạt trong Linux, cho phép quản lý quyền truy cập và xác thực người dùng một cách mềm dẻo mà không cần sửa đổi ứng dụng.
* Cách hoạt động của **PAM**:
  + Khi một dịch vụ hoặc ứng dụng yêu cầu xác thực (SSH, sudo, login...), nó gọi các mô-đun PAM.
  + PAM kiểm tra thông tin xác thực theo các quy tắc trong tệp cấu hình.
  + Nếu xác thực thành công, người dùng được cấp quyền truy cập.
* Cấu trúc của **PAM**: /etc/pam.conf hoặc thư mục /etc/pam.d chứa các tệp cấu hình cho từng dịch vụ như /etc/pam.d/sshd cho sshd . Có 4 loại modules:
  + **account** – Kiểm tra quyền truy cập (hạn chế thời gian, nhóm, tài khoản bị khóa).
  + **authentication** – Xác thực người dùng (mật khẩu, vân tay, khóa công khai).
  + **password** – Quản lý mật khẩu (đổi mật khẩu, kiểm tra độ mạnh).
  + **session** – Thiết lập môi trường sau khi đăng nhập (ghi log, giới hạn tài nguyên).

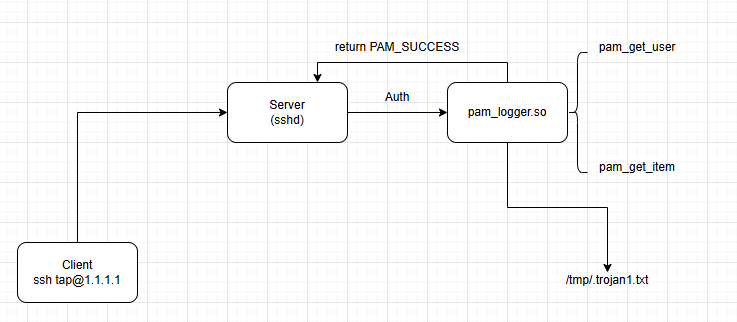
## Tìm hiểu cách hoạt động của **strace** và **ptrace**

* **ptrace** (Process Trace): là một syscall trong Linux cho phép tiến trình này giám sát và điều khiển tiến trình khác, thường dùng để debug. Các tác vụ chính:
  + Theo dõi hệ thống gọi (syscall) của tiến trình.
  + Đọc/ghi bộ nhớ tiến trình bị giám sát.
  + Kiểm soát thực thi (tạm dừng, tiếp tục, thay đổi luồng chạy).
* **strace** sử dụng **ptrace** để theo dõi và ghi lại các lệnh syscall mà tiến trình thực hiện, giúp debug lỗi và phân tích hiệu suất. Các tác vụ chính:
  + Hiển thị mọi syscall mà một chương trình thực hiện.
  + Theo dõi các tệp mà chương trình truy cập.
  + Đo thời gian thực thi của từng syscall.

# Thực hành

## Exercise 1

* **Ý tưởng**: Viết 1 module PAM và config nó để **sshd** dùng để xác thực



* **Giải thích**: Khi client ssh đến 1 server thì , sshd ( ssh deamon) của server sẽ tiến hành các bước xác thực danh tính, mật khẩu. Chúng ta sẽ config để sshd gọi đến 1 module xác thực (pam) mà chúng ta tự viết. Khi đó chúng ta sẽ viết lại các hàm của 1 module pam để log lại username và mật khẩu.

## Exercise 2

* **Ý tưởng**: Dùng **LD\_PRELOAD HOOKING** cho hàm **open()** và **read()** để lấy mật khẩu

Ảnh có chứa ảnh chụp màn hình, văn bản, hàng, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* **Giải thích**:
  + Đầu tiên sẽ đổi tên của bin **/usr/bin/ssh** thành **/user/bin/real\_ssh**
  + Sau đó sẽ tạo 1 file **/usr/bin/ssh** có nội dung là :
    - Echo “$@” > /tmp/.log\_sshtrojan2.txt && LD\_PRELOAD=/home/taptap/sshtrojan2.so /usr/bin/real\_ssh
    - Shell script này sẽ lưu tên username và tên server mà người dùng định ssh đến vào file log và chạy bin ssh thật
  + Biến **LD\_PRELOAD** là một biến môi trường được sử dụng để chỉ định một thư viện chia sẻ (shared library, thường có đuôi .so) sẽ được tải trước (preloaded) vào bộ nhớ trước khi chương trình chính thực thi
  + Điều này cho phép thay thế hoặc ghi đè các hàm trong thư viện tiêu chuẩn hoặc các thư viện khác mà chương trình sử dụng, mà không cần sửa đổi mã nguồn của chương trình và chúng ta sẽ ghi đề 2 hàm trong thư viện chuẩn là **open()** và **read()**
  + Syscall **open()** có tác dụng để mở 1 file và trả về **fd** (file descriptor), là một số nguyên duy nhất đại diện cho tệp đó trong quá trình thực thi chương trình.
  + Syscall **read()** có tác dụng đọc 1 file mà đã được mở bằng **open()**
  + Sau đó chúng ta sẽ viết lại 2 hàm trên theo ý tưởng: nếu thấy open **/dev/tty** thì chúng ta sẽ lưu lại **fd** , sau đó khi thấy hàm open đọc file có cùng **fd** đã lưu, tức là đọc dữ liệu từ terminal => lưu log lại.