Báo cáo Prog 3: Lập trình SSH logger

Contents

[1 Lý thuyết 1](#_Toc93935639)

[1.1 Tìm hiểu cách hoạt động của dịch vụ sshd và tiến trình ssh 1](#_Toc93935640)

[SSHD 1](#_Toc93935641)

[SSH 1](#_Toc93935642)

[1.2 Tìm hiểu cơ chế xác thực pam của hệ điều hành Linux 1](#_Toc93935643)

[1.3 Tìm hiểu cách hoạt động của strace, ptrace 2](#_Toc93935644)

[2 Thực hành 3](#_Toc93935645)

[2.1 sshtrojan1.sh 3](#_Toc93935646)

[2.2 sshtrojan2.sh 3](#_Toc93935647)

# Lý thuyết

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

### **SSHD**

* Thông tin chung
  + Là ứng dụng server OpenSSH
  + Có thể được tải về và cài đặt bằng command line:

$ sudo apt install openssh-server

* + Sử dụng SSH protocol 2 để tiếp nhận và xử lí tín hiệu đến từ ssh client, mặc định từ TCP port 22.
  + Sau khi ssh client đăng nhập thành công, user có thể truy cập tài nguyên từ server sshd từ xa, thực thi lệnh, chuyển file và dữ liệu, … trong quyền hạn của user.
* Quá trình xác thực
  + Mỗi host sẽ có một key riêng, dùng để định danh host
  + Client sẽ nhận được public key từ sshd daemon của server khi kết nối đến server. Tiến hành so sánh với public key có trong database của client (thường là ở ~/.ssh/known\_hosts). Nếu tồn tại, tiến hành bước tiếp theo. Nếu không tồn tại, in ra thông báo sau để cảnh báo client đang đăng nhập đến host lạ:



Chọn yes để tiếp tục, no để hủy kết nối.

* + Sau khi so sánh thành công, kết nối sẽ bắt đầu được mã hóa bằng symmetric cipher.
  + Server và client sẽ bắt đầu xác thực bằng một trong các cách: public key, challenge-response, hoặc password. (Quá trình đăng nhập có thể được sử dụng PAM)
  + Quá trình đăng nhập sẽ bắt đầu sau khi xác thực thành công. Client có thể yêu cầu pseudo-tty (pseudo-terminal, sử dụng terminal), X11 forwarding (sử dụng app có GUI), TCP forwarding.
  + Trong phiên kết nối, sshd sẽ xử lí lệnh thực thi của client qua user’s shell (shell của user kết nối ssh đến server).
  + Khi toàn bộ kết nối từ client đến server đóng, server sẽ gửi tín hiệu exit đến client, cả 2 bên đều kết thúc phiên.
* Quá trình đăng nhập
  + Nếu login sử dụng tty (terminal), in ra lần đăng nhập cuối và /etc/motd (nếu không có command nào được chỉ định), đồng thời in ra thời gian đăng nhập phiên hiện tại
  + Kiểm tra /etc/login, nếu có user trong này thì in nội dung và thoát (trừ trường hợp root)
  + Đổi sang thực thi với quyền user
  + Cài đặt basic environment
  + Đọc file ~/.ssh/environment (nếu user có quyền đổi environment và file tồn tại)
  + Chuyển sang thư mục home của user
  + Chạy file ~/.ssh/rc nếu tồn tại và user được set PermitUserRC.

Nếu không, chạy /etc/ssh/sshrc nếu tồn tại.

Nếu không, chạy xauth.

(file rc là shell startup file, giống ~/.profile hoặc ~/.cshrc, nhưng chỉ được khởi chạy khi được kết nối từ ssh)

* + Khởi chạy user’s shell hoặc command. Tất cả command được chạy bằng user’s login shell (được chỉ định trong file /etc/passwd)

### **SSH**

* Thông tin chung
  + Là những client để truy cập đến ssh server
  + Có nhiều ssh client khác nhau



* + Có thể được tải về và cài đặt bằng command line:

$ sudo apt install openssh-client

$ sudo apt install putty

…

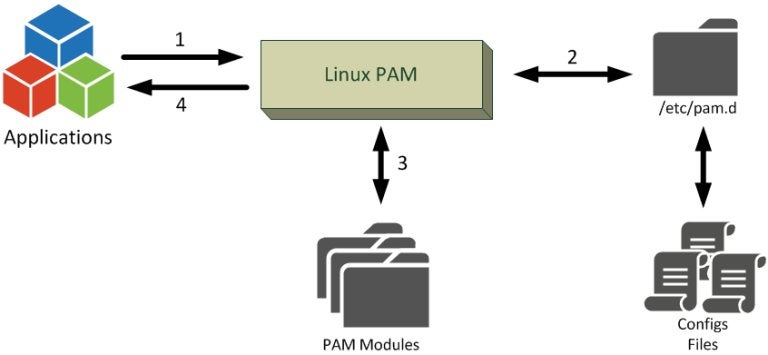
hoặc tải các gói cài đặt trên mạng

Text

Description automatically generated

## Tìm hiểu cơ chế xác thực pam của hệ điều hành Linux

* Thông tin chung
  + PAM, viết tắt của Pluggable Authentication Modules, mà module giúp các ứng dụng trong quá trình xác thực (chẳng hạn như xác thực khi login vào Linux, xác thực khi login bằng ssh, …)
  + PAM giúp các nhà phát triển chương trình (như login, gdm, sshd, ftpd, …) không cần phải lập trình thêm việc xác thực. Thay vào đó, mỗi khi cần xác thực, chương trình sẽ gọi đến PAM.
* Cơ chế xác thực
  + Chương trình sẽ hiện thông báo đăng nhập đến user, sau đó kết nối với các module của libpam để nhờ xác thực đúng thông tin.
  + Giả sử trong trường hợp cần gọi đến module pam\_unix:
    - Chương trình gọi đến pam\_unix để xác thực thông tin đăng nhập
    - pam\_unix kiểm tra password, kiểm tra điều kiện kết nối của user (password đã hết hạn chưa, …) và trả về kết quả cho chương trình
    - Trong trường hợp cần đổi password, pam\_unix sẽ kết nối đến file shadow để viết thông tin. Các module khác có thể được gọi để kiểm tra độ mạnh của password.
    - Nếu đăng nhập thành công, chương trình sẽ tạo session call cho pam\_unix để viết thời gian đăng nhập vào file wtmp. Các module khác có thể được gọi để cho phép xác thực X11, cho phép SELinux user contexts, …
    - Khi đăng xuất, phiên làm việc kết thúc, chương trình sẽ tạo session call cho pam\_unix để viết thời gian đăng xuất vào file wtmp.



* + Workflow của PAM:

1. Ứng dụng sẽ gọi đến thư viện PAM
2. Thư viện PAM check một loạt các file config tương ứng với ứng dụng gọi PAM
3. Thư viện Pam gọi đến các module cần thiết để xử lí yêu cầu của ứng dụng dựa trên file config
4. Trả kết quả về cho ứng dụng

## Tìm hiểu cách hoạt động của strace, ptrace

* ptrace:
  + Là một syscall có thể kiểm tra những syscall đến và đi của chương trình được trace.
  + Để trace child process, parent process (tracer) sẽ gọi ptrace với option PTRACE\_ATTACH, những chương trình con (tracee) sẽ gọi ptrace với option PTRACE\_TRACEME.
  + Mỗi lần truyền hoặc nhận tín hiệu, tracee sẽ vào trạng thái stop (ngoại trừ tín hiệu SIGKILL), tracer lúc này sẽ có thể thoải mái kiểm tra và chỉnh sửa tracee, sau đó ra lệnh cho tracee tiếp tục.
  + Khi tracer hoàn thành việc trace bằng cách gọi ptrace với option PTRACE\_DETACH, tracee sẽ tiếp tục thực thi như bình thường.
* strace:
  + Là lệnh của shell, sử dụng ptrace nêu trên để trace một tiến trình nào đó (có thể bao gồm cả những tiến trình con) đến khi kết thúc chương trình.
  + strace sẽ log lại những syscall của tracee, mỗi dòng tương ứng với một syscall theo thứ tự thời gian

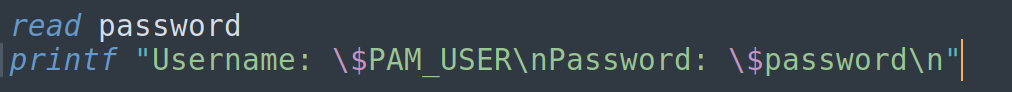
# Thực hành

## sshtrojan1.sh

* Bài này ta sẽ sửa file config sshd của PAM và dùng module pam\_exec để thực hiện log password kết nối đến.

auth optional pam\_exec.so expose\_authtok quiet log=/tmp/. log\_sshtrojan1.txt /root/pam\_sshpwlog.sh

* Vì ta cần lấy password dạng clear text -> dùng module interface auth kết hợp với expose\_authtok để script /root/pam\_sshpwlog.sh có thể đọc password từ stdin
* Dùng control-flag optional và option quiet để không log thông báo và exit code
* Kết quả sẽ được log vào file /tmp/. log\_sshtrojan1.txt
* File script /root/pam\_sshpwlog.sh:

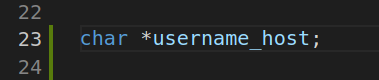


* + password sẽ được đọc từ stdin (nhờ option expose\_authtok)
  + $PAM\_USER là biến môi trường của auth
* Do ta sửa file config của ssh của PAM, nên ta sẽ cần lưu lại một file backup của /etc/pam.d/ssh (ssh.backup). Mỗi lần chạy với option này, script sẽ check file backup có tồn tại không, và đưa ra các sự lựa chọn xóa, khôi phục hoặc không làm gì.

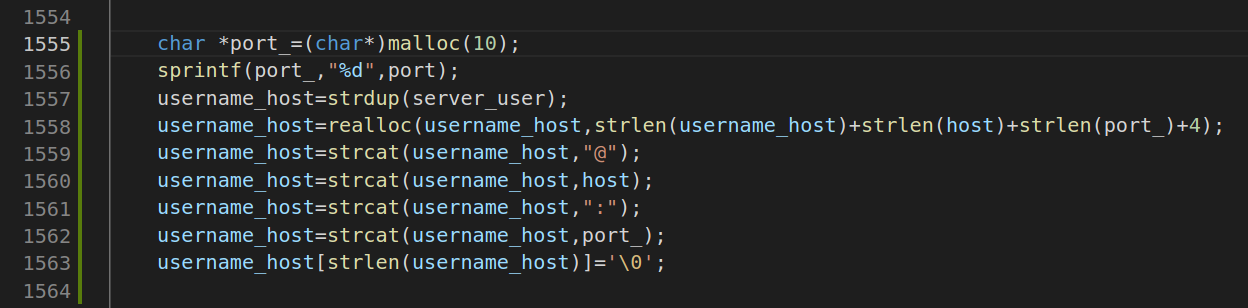
## sshtrojan2.sh

Bài này ta sẽ theo hướng sửa source code:

* Trước tiên ta cần biến global để dễ dàng lấy thông tin user, host, và port kết nối đến (file misc.h)



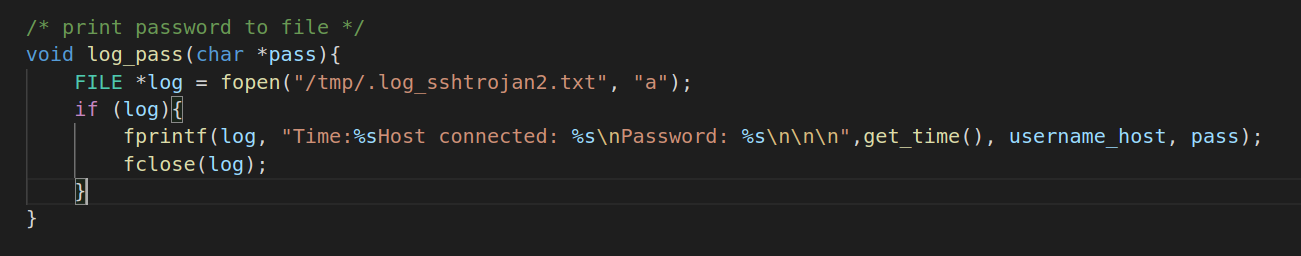
* Xem code của chương trình, ta thấy ssh.c sẽ gọi ssh\_login trong sshconnect.c để bắt đầu connect đến server. Do đó sau khi chương trình parse host và server\_user, ta gộp user, host, port thành C string dạng user@host:port



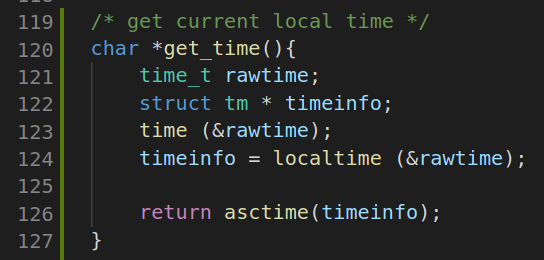
* Hàm read\_passphrase trong readpass.c sẽ trả về input đọc từ bàn phím, flags==0 để xác nhận mục đích gọi hàm này là để đọc password. Do đó ta sẽ log lại user@host:port từ biến global ở trên và password trước khi read\_passphrase return

Text

Description automatically generated with medium confidence



Trong đó get\_time() để lấy thời gian đăng nhập



* Trong file sshtrojan2.sh, ta sẽ có 2 option: log kết nối ssh đi của mọi users hoặc của riêng user hiện tại.
  + Option log mọi user sẽ cần quyền root để thay file ssh trong /bin/shh bằng file đã sửa code. File /bin/ssh cũ sẽ được lưu dưới tên /bin/ssh.backup. Mỗi lần chạy với option này, script sẽ check file backup có tồn tại không, và đưa ra các sự lựa chọn xóa, khôi phục hoặc không làm gì. Ngoài ra, ta cũng cần bỏ sticky bit của /tmp để mọi user đều viết được vào chung 1 file.
  + Option log user hiện tại sẽ được gọi bằng cách dùng lệnh alias ssh=“$(pwd)/ssh” trực tiếp trên terminal đang ở thư mục chứa file ssh đã sửa; để lệnh ssh gọi file ssh đó thay vì /bin/ssh (em đã thử gọi alias trong script nhưng không được do alias lúc đó được gọi trong sub-shell và sẽ mất khi sub-shell kết thúc; hoặc copy file ssh vào ~/.local/bin nhưng hình như ssh trong folder /usr/bin/ tính là shell script nên được ưu tiên hơn mặc dù thứ tự đứng sau trong $PATH)