**PROG03: LẬP TRÌNH SSH LOGGER**

Mục lục

[**A.** **Lý thuyết** 1](#_Toc158025212)

[**1.** **Tìm hiểu cách hoạt động của dịch vụ sshd và tiến trình ssh.** 1](#_Toc158025213)

[**2.** **Tìm hiểu cơ chế xác thực pam của hệ điều hành linux** 2](#_Toc158025214)

[**3.** **Tìm hiểu cách hoạt động của strace, ptrace** 4](#_Toc158025215)

[**B.** **Thực hành** 6](#_Toc158025216)

[**1.** **Sshtrojan1 - Một user tiến hành truy cập vào máy tính đó thông qua ssh** 6](#_Toc158025217)

[**2.** **Sshtrojan2 - Một người dùng đứng trên máy tính đó ssh đi một máy tính khác** 8](#_Toc158025218)

[**C.** **Tài liệu tham khảo** 10](#_Toc158025219)

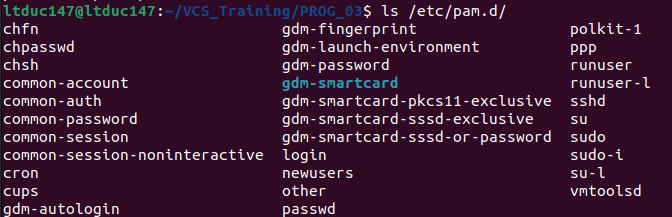
1. **Lý thuyết**
2. **Tìm hiểu cách hoạt động của dịch vụ sshd và tiến trình ssh.**

* SSH là một giao thức mạng cho phép người dùng truy cập từ xa vào máy tính khác một cách an toàn. Nó hoạt động bằng cách mã hóa tất cả dữ liệu được truyền giữa máy khách và máy chủ, ngăn chặn kẻ tấn công nghe trộm hoặc giả mạo kết nối.
* Dịch vụ sshd: là tiến trình nền chạy trên máy chủ, lắng nghe các kết nối SSH từ máy khách trên một cổng nhất định (mặc định là 22). Khi có một máy khách gửi yêu cầu kết nối, sshd sẽ sinh ra một tiến trình nền để tiến hành xử lý trao đổi khóa, mã hóa, xác thực, thực thi lệnh và trao đổi dữ liệu:
  + Xác thực người dùng sử dụng mật khẩu hoặc khóa SSH (cặp khóa công khai/bí mật để xác minh danh tính).
  + Thiết lập kết nối an toàn sử dụng mã hóa đối xứng và bất đối xứng.
  + Mở một shell hoặc ứng dụng khác trên máy chủ. Điều này cho phép người dùng tương tác với máy chủ từ xa.
* Tiến tình ssh: chạy trên máy khách và thực hiện yêu cầu kết nối SSH với máy chủ (để đăng nhập và thực hiện các lệnh trên máy từ xa). Khi một người dùng chạy lệnh ssh, tiến trình ssh sẽ thực hiện các bước sau:
  + Kết nối với máy chủ sử dụng địa chỉ IP hoặc tên máy chủ của máy chủ.
  + Xác thực người dùng sử dụng mật khẩu hoặc khóa SSH (cặp khóa công khai/bí mật để xác minh danh tính).
  + Thiết lập kết nối an toàn sử dụng mã hóa đối xứng và bất đối xứng.
  + Tương tác từ xa với máy chủ thông qua shell được mở ở máy chủ.
* Quy trình khi một máy khách tạo kết nối đến máy chủ từ xa sử dụng ssh
  + Client bắt đầu bằng việc gửi ID tương ứng với cặp khoá private – public key mà nó muốn xác thực lên server
  + Server sẽ kiểm tra và tìm kiếm một public tương ứng với ID mà client đã gửi lên
  + Nếu một public key được tìm thấy, server sẽ generate một số ngẫu nhiên và sử dụng public key để mã hoá con số này.
  + Server tiến hành gửi cho client một message đã được mã hoá bởi public key.
  + Client nhận message từ server và sử dụng private key để giải mã message và nhận lại con số ban đầu được server sinh ra.
  + Client sẽ sử dụng con số được giải mã kết hợp với shared session key đang được sử dụng để giao tiếp và sử dụng MD5 để tính toán giá trị băm cho chúng.
  + Client gửi mã hash MD5 được tính toán ở trên cho server như một câu trả lời cho con số đã được mã hoá trước đó bởi server.
  + Server sử dụng cùng shared session key và con số mà nó đã generate ra để tính ra một mã hash MD5 đem đi so sánh với mã hash mà client gửi lên. Nếu 2 giá trị này trùng khớp thì một kết nối an toàn được thiết lập.

1. **Tìm hiểu cơ chế xác thực pam của hệ điều hành linux**

* PAM (viết Pluggable Authentication Modules), là một framework cung cấp một hệ thống xác thực linh hoạt. PAM tách quy trình xác thực thành một loạt mô-đun độc lập, cho phép quản trị viên hệ thống định cấu hình và tùy chỉnh các chính sách xác thực mà không cần sửa đổi mã nguồn của ứng dụng.
* PAM hoạt động thông qua một loạt các mô-đun xếp chồng lên nhau, mỗi mô-đun thực hiện một nhiệm vụ xác thực cụ thể. Các mô-đun này hoạt động cùng nhau để xác minh danh tính của người dùng trước khi cấp quyền truy cập vào hệ thống hoặc dịch vụ. Quy trình làm việc chung bao gồm:
  + Ứng dụng khởi tạo PAM: Khi người dùng cố gắng truy cập một ứng dụng nhận biết PAM (ví dụ: login, SSH), ứng dụng sẽ kích hoạt PAM để bắt đầu quá trình xác thực.
  + Dựa trên ứng dụng và dịch vụ cụ thể được yêu cầu, PAM tải một chuỗi các mô-đun có liên quan từ các thư viện được lưu trữ bên trong các thư mục như /lib hoặc /usr/lib (Các mô-đun liên quan này được cấu hình trong thư mục /etc/pam.d/ cho từng ứng dụng).

**PAM config:**



**Lib PAM module:**

A computer screen shot of a program

Description automatically generated

Mỗi mô-đun thực hiện một chức năng cụ thể, chẳng hạn như:

* + - Auth modules: xác thực xác minh danh tính của người dùng, ví dụ bằng cách yêu cầu và kiểm tra mật khẩu hoặc bí mật khác.
    - Account modules: Kiểm tra xem tài khoản được chỉ định có phải là mục tiêu xác thực hợp lệ trong các điều kiện hiện tại hay không. Điều này có thể bao gồm các điều kiện như hết hạn tài khoản, thời gian trong ngày và người dùng có quyền truy cập vào dịch vụ được yêu cầu.
    - Password modules: chịu trách nhiệm cập nhật mật khẩu và thường được kết hợp với các mô-đun được sử dụng trong bước xác thực.
    - Session module: xác định các hành động được thực hiện vào đầu và cuối phiên. Một phiên bắt đầu sau khi người dùng xác thực thành công.
  + Các mô-đun PAM tương tác với nhau theo thứ tự xếp chồng lên nhau được chỉ định trong tệp cấu hình PAM, chuyển kết quả xác thực và thông tin trạng thái từ mô-đun này sang mô-đun tiếp theo. Thứ tự thực hiện điển hình là:

**‘auth’ -> ‘account’ -> ‘password’ -> ‘session’**

Mỗi mô-đun có thể có các cờ kiểm soát như ‘**required’**, ‘**requisite’**, ‘**sufficient’, ‘optional’,** đểxác định tác động của sự thành công hay thất bại của mô-đun đối với quá trình xác thực tổng thể.

**Vd PAM module flag:**

A computer screen with white text

Description automatically generated

* + Kết quả xác thực tổng thể được xác định dựa trên kết quả tổng hợp của tất cả các mô-đun. Nếu tất cả các mô-đun thành công, quyền truy cập sẽ được cấp. Nếu bất kỳ mô-đun nào bị lỗi, quyền truy cập sẽ bị từ chối và người dùng có thể được nhắc cung cấp thông tin bổ sung hoặc nhận được thông báo lỗi.

1. **Tìm hiểu cách hoạt động của strace, ptrace**

* **ptrace:** lệnh gọi hệ thống (system call) được cung cấp bởi linux kernel, cho phép tiến trình cha quan sát và kiểm soát việc thực thi của tiến trình con.
  + Sử dụng cho mục đích gỡ lỗi và truy vết
  + Ptrace cho phép tiến trình cha:
    - Đính kèm với tiến trình con
    - Chạy từng câu lệnh một
    - Kiểm tra và sửa đổi bộ nhớ các thanh ghi của tiến trình con
    - Nhận thông báo khi xảy ra sự kiện nhất định, như tín hiệu hoặc quá trình chấm dứt
  + Một số bước chung để sử dụng ptrace
    - Tiến trình cha sử dụng fork() để tạo tiến trình con
    - Tiến trình cha (tracer) có thể bắt đầu trace tiến trình con (tracee) bằng cách sử dụng lệnh ptrace() với option PTRACE\_ATTACH, hoặc tiến trình con sẽ gọi ptrace() với option PTRACE\_TRACEME**.**
    - Trong quá trình bị trace, tracee sẽ dừng lại (trạng thái STOP) mỗi khi nhận được tín hiệu (ngoại trừ SIGKILL), tracer sẽ được thông báo ở lần gọi waitpid() tiếp theo. Trong khi tracee bị dừng lại, tracer có thể thực hiện nhiều hành động debug hoặc trace sử dụng lệnh ptrace, và sau đó ra lệnh cho tracee tiếp tục
    - Tiến trình cha kết thúc việc trace bằng việc tách khỏi tiến trình con sử dụng lệnh ptrace() với option PTRACE\_DETACH
* **strace:** tiện ích có chức năng chặn và ghi lại các lệnh gọi hệ thống mà chương trình thực hiện trong quá trình thực thi.
  + Khi bạn chạy một chương trình có strace, nó sẽ gắn vào tiến trình đích và giám sát các lệnh gọi hệ thống được thực hiện bởi tiến trình đó (được thực hiện thông qua ptrace).
  + Strace tiến hành in ra chi tiết từng lệnh gọi hệ thống (system call), bao gồm số lượng, tham số, giá trị trả về.

**A computer screen with white and blue text

Description automatically generated**

* + Hoạt động cơ bản của strace:
    - Gắn vào tiến trình đích sử dụng ptrace
    - Khiến tiến trình đích dừng lại ở mỗi system call
    - Kiểm tra và in thông tin về system call
    - Cho phép tiến trình đích tiếp tục thực thi cho đến khi có lệnh gọi system call tiếp theo
    - Lặp lại cho đến khi tiến trình đích hoàn thành hoặc được tách ra

1. **Thực hành**

Trường hợp đang có quyền root để truy cập vào 1 máy tính linux. Xây dựng chương trình để lấy được username, password ssh trong 2 trường hợp sau mà không thay đổi luồng hoạt động bình thường:

1. **Sshtrojan1 - Một user tiến hành truy cập vào máy tính đó thông qua ssh**

**Ý tưởng:** sử dụng strace lên dịch vụ sshd để trace các kết nối ssh đến máy nạn nhân.

* Thử sử dụng strace để xem các system call được sinh ra khi có kết nối ssh đến máy chủ. Sử dụng câu lệnh sau:

**sudo strace -qfvx -s 1024 -p $(pgrep -o sshd) -o strace.log**

**Phân tích strace log:**

* Khi người dùng nhập sai mật khẩu, sshd sẽ gọi system call sendto để ghi dòng sau vào syslog:

**sendto(5, "<38>Feb 5 09:12:37 sshd[809387]: Failed password for UserA from ….**

* Khi người dùng nhập mật khẩu đúng, sshd sẽ gọi system call sendto để ghi dòng sau vào syslog:

**sendto(5, "<38>Feb 5 09:12:55 sshd[809387]: Accepted password for UserA from ….**

* Từ dấu hiệu này ta có thể biết được user có nhập mật khẩu đúng hay không và lấy username từ dòng này, sau đó lấy lần nhập mật khẩu cuối cùng làm mật khẩu để ghi vào /tmp/.log\_sshtrojan1.txt
* Để tìm cách lấy mật khẩu, ta phân tích 3 lần nhập mật khẩu trong một kết nối ssh.

**A screen shot of a computer code

Description automatically generated**

**A computer screen shot of numbers and equations

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

* Sshd sẽ fork một tiến trình con dành cho việc ghi mật khẩu và một tiến trình con dành cho việc đọc mật khẩu.
* (1) Trước khi ghi mật khẩu, tiến trình con sẽ dùng system call write để ghi trước 5 byte dữ liệu và file descriptor 5 (giá trị này có thể khác nhau tùy thuộc vào config), bao gồm:
  + \x00\x00\x00: 3 byte rỗng
  + \x0e: tương ứng với độ dài mật khẩu cộng thêm 5
  + \x0c: biểu diễn hex của ký tự form feed \f
* (2) Tiếp theo, tiến trình con tiến hành ghi tiếp vào fd 5 các dữ liệu sau:
  + \x00\x00\x00: 3 byte rỗng
  + \x09: độ dài mật khẩu
  + \x30\x35…\x31: giá trị mật khẩu thật sự

Từ (1) ta có thể truy ra vị trí của (2) và dùng (2) để lấy được mật khẩu của người dùng.

**Giải thích chương trình:**

* Đầu tiên, chương trình sẽ dùng lệnh pgrep để tìm kiếm pid của sshd service, sau đó strace lên dịch vụ này (chỉ trace 2 system call write và sendto theo phân tích ở trên) và đưa strace log ra file /tmp/.strace\_sshd.log
* Chương trình sẽ liên tục kiểm tra file strace log xem đã xuất hiện dòng chữ “Accepted password” (đã nhập đúng mật khẩu) hoặc “PAM 2 more authentication failures” (đã nhập sai 3 lần) để tiến hành gọi hàm parse\_credentials() để tiến hành lấy mật khẩu
* Hàm parse\_credentials()
  + Kiểm tra dấu hiệu người dùng nhập mật khẩu đúng
  + Tiến hành lấy username từ dữ liệu được ghi vào syslog bằng systemcall sendto (**sendto(5, "<38>Feb 5 09:12:55 sshd[809387]: Accepted password for UserA …)**
  + Lọc trong strace log dấu hiệu của việc ghi mật khẩu (tương ứng với **(1)** ở phân tích trên: **write(5, "\x00\x00\x00\x0e\x0c", 5) = 5**) và lấy systemcall write (tương ứng với **(2)** ở phân tích trên : **write(5, "\x00\x00\x00\x09\x30\x35\x36\x34\x37\x32\x38\x33\x31", 13 ….**) ngay sau nó (option -A 1), lấy lần nhập cuối cùng (tail -n 1) và trích xuất dữ liệu nằm trong cặp ngoặc kép **""** (cut -d '"' -f2) của **(2)** và bỏ đi 4 byte đầu (cut -c 17-) để lấy được mật khẩu thực sự (ở dạng hex)
  + Ghi username và password vào file /tmp/.log\_sshtrojan1.txt
  + Kill tiến trình strace để chuẩn bị cho kết nối mới

1. **Sshtrojan2 - Một người dùng đứng trên máy tính đó ssh đi một máy tính khác**

**Ý tưởng:** sử dụng strace lên tiến trình ssh đang chạy để đọc mật khẩu của người dùng khi kết nối đến máy khác.

* Thử sử dụng strace để xem các system call được sinh ra khi 1 người dùng chạy tiến trình ssh:

**sudo strace -fv -p $(ps aux | grep -w ssh | grep @ | awk '{print $2}')**

A screenshot of a computer

Description automatically generated

* Ta có thể thấy tiến trình ssh sẽ gọi system call read để chờ người dùng nhập mật khẩu và đọc từng ký tự một và kết thúc khi gặp kí tự xuống dòng (\n)

**Giải thích chương trình:**

* Đầu tiên, chương trình sẽ tìm kiếm pid của các tiến trình ssh đang chạy, nếu tồn tại thì sẽ dùng strace để trace các pid này (chỉ trace system call read)
* username sẽ được lấy từ câu lệnh ssh của người dùng (được hiển thị khi dùng ps)
* Sau đó tiến hành đọc từng dòng của strace log, nếu chưa gặp kí tự xuống dòng thì thêm lần lượt các kí tự được nhập vào biến password, ngược lại ghi username và password vào file /tmp/.log\_sshtrojan2.txt và kết thúc
* Tiếp tục với các lần ssh tiếp theo

**Link video demo của 2 chương trình:** [duclt22\_prog03.mp4 - Google Drive](https://drive.google.com/file/d/1PddsgYb-DIg6LXT9MvwC9BwM8X0LyZZI/view)

**Link mã nguồn 2 chương trình:** [VCS\_Training/PROG\_03 at master · ltduc147/VCS\_Training (github.com)](https://github.com/ltduc147/VCS_Training/tree/master/PROG_03)

1. **Tài liệu tham khảo**

**A.1**

[sshd(8) - OpenBSD manual pages](https://man.openbsd.org/sshd.8)

[ssh(1) - OpenBSD manual pages](https://man.openbsd.org/ssh.1)

[SSH là gì? Cơ chế hoạt động của SSH - Deft Blog (shareprogramming.net)](https://shareprogramming.net/ssh-la-gi-co-che-hoat-dong-cua-ssh/)

[SSH hoạt động như thế nào? (viblo.asia)](https://viblo.asia/p/ssh-hoat-dong-nhu-the-nao-Eb85oJEjl2G)

**A.2**

[Linux PAM - Wikipedia](https://en.wikipedia.org/wiki/Linux_PAM)

[Understanding PAM Authentication and Security (aplawrence.com)](https://aplawrence.com/Basics/understandingpam.html)

[Pluggable Authentication Modules - IBM Documentation](https://www.ibm.com/docs/en/aix/7.2?topic=system-pluggable-authentication-modules)

**A.3**

[Looking into Process Tracing and Control in Linux with strace and ptrace | by Razika Bengana | Medium](https://medium.com/@razika28/looking-into-process-tracing-and-control-in-linux-with-strace-and-ptrace-f335330d1500)

[ptrace(2) - Linux manual page (man7.org)](https://man7.org/linux/man-pages/man2/ptrace.2.html)

[How does strace work? | Packagecloud Blog](https://blog.packagecloud.io/how-does-strace-work/)

**B.1,2**

[Strace Spelunking: Diving Deep into SSH Password Discovery | Home (kevinsuckiel.io)](https://www.kevinsuckiel.io/Strace-Spelunking/)

[SSH Password Theft · Eapolsniper's Blog](https://eapolsniper.github.io/2020/08/22/SSH-Password-Theft/)

[Capturing ssh passwords on host using root privileges - Notes\_Wiki (sbarjatiya.com)](https://www.sbarjatiya.com/notes_wiki/index.php/Capturing_ssh_passwords_on_host_using_root_privileges)