Lý thuyết

-Cách hoạt động của dịch vụ sshd

- Quá trình xác thực của dịch vụ sshd

+ Bất kì khi nào 1 người dùng kết nối, daemon sẽ phản hồi bằng các public host và server key

+ Người dùng sẽ so sánh RSA host key với cơ sở dữ liệu của người dùng để xác minh rằng key không thay đổi

+ Sau đó người dùng tạo 1 số ngẫu nhiên 256-bit

+ Daemon sẽ mã hóa số ngẫu nhiên này làm session key để mã hóa tất cả các giao thức khác trong session

+ Cuối cùng máy chủ và người dùng sẽ đi vào bước authentication dialog khi người dùng sẽ tự xác thực bản thân bằng public key authentication, password authentication

+ Tài khoản có thể đăng nhập không thành công , không được phép truy cập nếu tài khoản nằm trong DenyUsers hoặc nhóm của nó nằm trong DenyGroup

* Quá trình đăng nhập của dịch vụ SSHD

+ Nếu đăng nhập trên tty và không có lệnh nào được thực thi, SSHD sẽ in thời gian đăng nhập cuối cùng và chạy lệnh /etc/motd

+ Nếu đăng nhập trên tty, ghi lại thời gian đăng nhập

+ kiểm tra tệp /etc/nologin, nếu tệp tin đó tồn tại, in ra nội dung và thoát

+ Thay đổi sang chế độ chạy với các đặc quyền của người dùng thông thường

+ Thiết lập môi trường cơ bản

+ Đọc tệp ~/.ssh/environment, nếu tệp tồn tại thì người dùng được phép thay đổi môi trường của họ

+ Chuyển đến thư mục chính của người dùng

+ Nếu ~/.ssh/rc tồn tại, chạy tệp. Nếu /etc/ssh/sshrc tồn tại ,chạy tệp.

+ Thực thi user’s shell hoặc command

* 3 bước của tiến trình SSH

+Bước 1: Xác minh máy chủ bởi người dùng

* Người dùng khởi tạo kết nối SSH với máy chủ
* Máy chủ listen port mặc định 22 cho các kết nối SSH
* Danh tính máy chủ được xác minh bằng 2 trường hợp
  + Th1: Nếu người dùng truy cập máy chủ lần đầu, máy người dùng sẽ được yêu cầu xác thực máy chủ theo cách thủ công bằng các xác minh public key của máy chủ
  + TH2: Nếu người dùng truy cập vào máy chủ không phải lần đầu tiên, danh tính máy chủ sẽ được xác minh với thông tin đã được ghi trước đó trong tệp known\_hosts

+ Bước 2: Tạo session key để mã hóa tất cả những thông tin liên lạc

* Sau khi máy chủ được xác minh, cả 2 bên sẽ tạo session key sử dụng thuật toán diffie hellman
* Session key được tạo là khóa đối xứng chung, tức là 1 khóa được sử dụng để mã hóa và giải mã

+ Bước 3: Xác thực người dùng

* Máy người dùng bắt đầu bằng cách gửi 1 ID cặp khóa mà nó muốn xác thực tới máy chủ
* Máy chủ kiểm tra tệp authorized\_keys của tài khoản mà máy người dùng đang cố gắng đăng nhập để tìm ID của khóa
* Nếu tìm thấy public key có ID giống trong tệp, máy chủ sẽ tạo 1 số ngẫu nhiên và sử dụng public key để mã hóa số và gửi tin nhắc được mã hóa
* Nếu máy người dùng có private key đúng, nó sẽ giải mã tin nhắn để lấy số ngẫu nhiên do máy chủ tạo ra
* Máy người dùng số ngẫu nhiên thu được với session key được chia sẻ để tính toán giá trị MD5
* Sau đó máy người dùng sẽ giử lại mã md5 này cho máy chủ như 1 câu trả lời cho số đã được mã hóa
* Máy chủ sử dụng cùng 1 session key được chia sẻ và số ban đầu mà nó đã gửi cho máy người dùng để tự tính toán giá trị MD5
* Máy chủ so sánh kết quả của nó với kết quả mà khách hàng đã gửi lại. Nếu 2 giá trị này khớp nhau, điều đó chứng tỏ rằng máy chủ người dùng đang sở hữu private key và người dùng được xác thực
* Cơ chế xác thực pam trong hệ điều hành linux
* Bước 1: Người dùng chạy một ứng dụng để truy cập vào dịch vụ mong muốn, ví dụ: login.
* Bước 2: PAM-aware application (chương trình giúp các privile-granting application làm việc với thư viện PAM) gọi thư viện PAM để thực hiện nhiệm vụ xác thực.
* Bước 3: PAM Library sẽ dựa vào file cấu hình của chương trình đó trong /etc/pam.d (ví dụ ở trên là dịch vụ login thì file cấu hình sẽ là /etc/pam.d/login ) để xác định loại xác thực nào được yêu cầu cho chương trình trên. Trong trường hợp không có file cấu hình, thì file /etc/pam.d/other sẽ được sử dụng.
* Bước 4: PAM Library sẽ load các module yêu cầu cho việc xác thực được đề cập ở trên.
* Bước 5: Các modules này sẽ tạo một liên kết tới các hàm chuyển đổi trên chương trình.
* Bước 6: Các hàm này dựa vào các modules mà đưa ra các yêu cầu với người dùng (ví dụ ở login, chúng có thể yêu cầu người dùng nhập password)
* Bước 7: Người dùng nhập thông tin vào theo yêu
* Bước 8: Sau khi quá trình xác thực kết thúc, chương trình này sẽ dựa vào kết quả mà đáp ứng yêu cầu người dùng (ví dụ cho phép login vào hệ thống) hay thông báo thất bại với người dùng nếu quá trình xác thực thất bại.
* Cách hoạt động của strace, ptrace.

+ strace: Khi một câu lệnh được thực thi, strace sẽ tạo ra 2 process một để thực hiện chương trình , một để trao theo dõi chương trình đó

+ Khi chương trình được theo dõi, nó sẽ thực hiện ptrace request lên Ptrace\_traceme

+ Tiến trình theo dõi sẽ bắt đầu theo dõi, được báo hiệu khi chương trình đang được thực thi gửi lại SIGSTOP

+ Khi này chương trình được theo dõi sẽ chạy đến hoặc điểm khởi đầu hoặc thoát khỏi system calls

+ Khi chương trình được theo dõi đến 1 system call, PTRACE\_PEEKUSER sẽ decode system call number