CÂU LẠC BỘ AN TOÀN THÔNG TIN PTIT

ISP CLUB

****

BÁO CÁO BÀI TẬP

WEEK 4: SSH LOGGER

Giảng viên hướng dẫn: **Mùa hè zui zẻ**

Tên sinh viên: **Nguyễn Anh Vũ**

Mã sinh viên: **B21DCAT224**

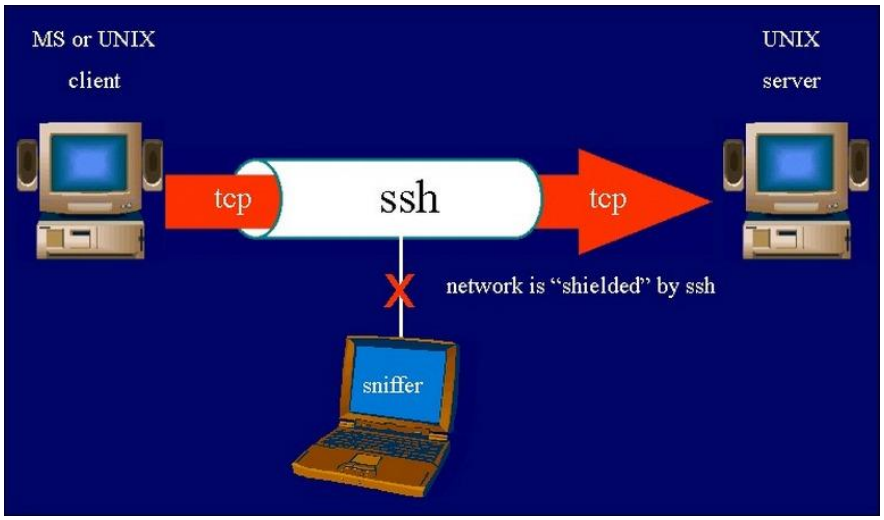
HÀ NỘI – 2022

MỤC LỤC

LÝ THUYẾT:

I. Cách hoạt động của dịch vụ sshd và tiến trình ssh:

- OpenSSH: **bộ các chương trình** giúp **mã hóa** các giao dịch giữa các hosts thông qua **SSH**
- SSH: **S**ecure **Sh**ell – giao thức (giống http, https, tcp, etc.) hỗ trợ **truy cập** vào **máy chủ** từ xa thông qua internet 1 cách **bảo mật** bằng cách **mã hóa dữ liệu** giao tiếp giữa 2 máy; dùng để thay thế Telnet, rlogin, rsh



- Cơ chế xác thực:
  + Dùng **mật khẩu**
  + Xác thực bằng **key pair**
- Hầu hết sử dụng trên **terminal/command line**
- *Điều kiện*: server phải có **SSHD** (OpenSSH Daemon) - một phần của quá trình triển khai OpenSSH
- Các kỹ thuật mã hóa:
  + **Symmetric** encryption (tên khác: **shared secret** encryption, **secret key** encryption): sử dụng **chung key** (hoặc **key pair** – có thể suy ra cái còn lại nếu biết 1 cái) để mã hóa và giải mã; thường dùng để **mã hóa toàn bộ liên lạc** trong phiên giao dịch; key được hình thành nhờ thuật toán **DHKE**

+ **Asymmetric** encryption: sử dụng **public key** để **mã hóa** và **private key** để **giải mã**; thường dùng để **trao đổi thuật toán** của symmetric encryption

+ **Hashing**: chỉ dùng để **mã hóa**, dùng để **xác nhận tính xác thực**

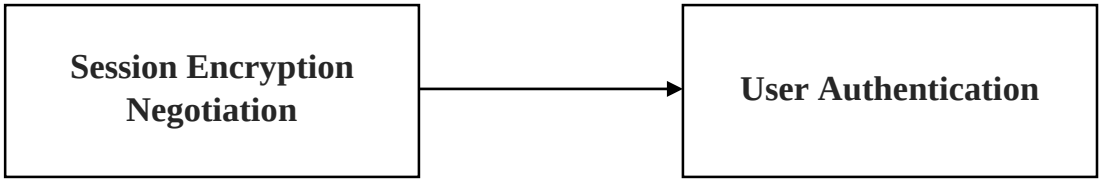
- Ứng dụng:

+ Chia sẻ file một cách bảo mật

+ Quản lý thiết bị và tài khoản từ xa

II. Cơ chế xác thực của Linux:

- Để **kết nối** với 1 server từ xa: dùng lệnh *ssh <username>@<ip/domain_name>*
- Cài đặt **cấu hình SSH** cho server: file */etc/ssh/sshd_config*
- Cơ chế hoạt động:
  + Dựa trên mô hình client-server
  + Có 2 giai đoạn thiết lập kết nối:



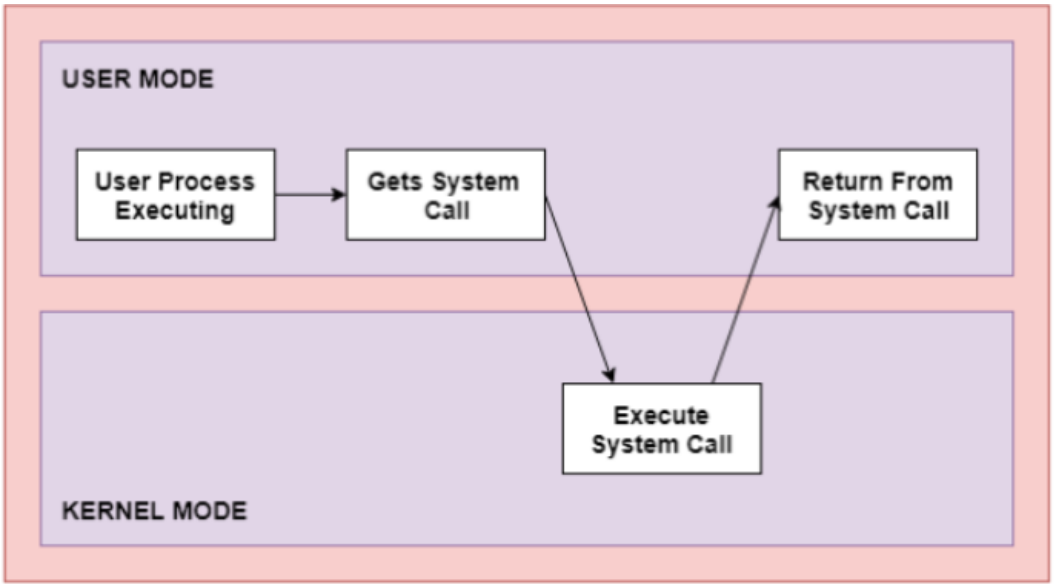
- *Session Encryption Negotiation*: khi client kết nối tới server qua TCP, server sẽ trình ra **encryption protocol version**; nếu client cũng có protocol **tương ứng**, server gửi symmetric public key để client xác thực tính chính xác của server; lúc này, hai bên "negotiate" tạo ra symmetric key bằng thuật toán **Diffie-Hellman Key Exchange (DHKE)**
- *User Authentication*: khi client được cấp quyền vào server, tùy vào server sẽ có cách xác thực riêng; phương pháp chung là **chứng thực bằng mật khẩu**, nhưng phương pháp phổ biến (và recommended) là dùng **SSH key pairs**

- Để tạo ra các key: dùng lệnh *ssh-keygen*
  + *~/.ssh/id_rsa* (private key): file mặc định chứa **private key**
  + *~/.ssh/id_rsa.pub* (public key): file mặc định chứa **public key**
  + Để chuyển public key sang server: dùng lệnh *ssh-copy-id <username>@<ip/domain_name>*
  + Public key bên server ở trong file *authorized_keys*

III. Cách hoạt động của strace, ptrace:

- strace (system trace): dùng để **theo dõi** các **system call** đến hệ thống của tiến trình

  ❗ *System call*: là cách mà một chương trình **yêu cầu một dịch vụ** từ **nhân** OS để tương tác với



+ Cách dùng:

- *strace -p <pid>*: theo dõi các system call của tiến trình *pid*
- *strace <command>*: theo dõi các system call mà command đó gọi
- *strace -o <text.txt> <command>*: lưu output của strace vào file *text.txt*
- *strace -c <command>*: output một bảng thông tin số lượng và tên system call tương ứng, kèm theo thời gian thực thi system call đấy

VD:

```
vux@vux:~/Music$ strace ls
execve("/usr/bin/ls", ["ls"], 0x7ffd7aa404e0 /* 55 vars */) = 0
brk(NULL)                               = 0x5615d3934000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe21fda710) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fda7418c000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=58633, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 58633, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fda7417d000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=166280, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 177672, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fda74151000
mprotect(0x7fda74157000, 139264, PROT_NONE) = 0
mmap(0x7fda74157000, 106496, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7fda74157000
mmap(0x7fda74171000, 28672, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x20000) = 0x7fda74171000
mmap(0x7fda74179000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x27000) = 0x7fda74179000
mmap(0x7fda7417b000, 5640, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fda7417b000
close(3)                                = 0
```

- ptrace (process trace): dùng để giúp một tiến trình (**tracer**) có thể **quản lý** một tiến trình khác (**tracee**)

❗ Kiến thức bổ sung:

1. PAM:

- *Chứng thực động* (Dynamic Authentication): sử dụng crypto để tạo ra sự chứng thực **khác nhau** cho **mỗi phiên** giao dịch
- **P**luggable **A**uthentication **M**odules: cơ chế tích hợp các **hệ thống xác nhận mức độ thấp** vào **API** (giao diện lập trình ứng dụng) mức độ cao (nói cách khác, là thư viện hỗ trợ các **chương trình** có **sự chứng thực người dùng** như *sshd* và *su*)
- Để deploy PAM, mỗi chương trình/ứng dụng phải là **PAM-ware**
- Để check xem một chương trình có phải PAM-ware hay không: dùng lệnh *ldd <tên ứng dụng>* (lệnh *ldd* dùng để list các dependency
- Tệp cấu hình của PAM: */etc/pam.conf* và */etc/pam.d/** (dir chứa cấu hình PAM cho các PAM-ware)
- Cú pháp trong file .conf:

service type control-flag module module-arguments

+ *service*: actual app name – tên dịch vụ, tên của chương trình/ứng dụng

+ *type*: module type/context/interface

Type	Ý nghĩa
account module	kiểm tra mật khẩu và tài khoản, sau khi được xác minh qua **auth** module, quyết định nếu người dùng này có quyền truy cập không
authentication module	xác thực người dùng
password module	cho phép **thay đổi mật khẩu**
session module	quản lý **phiên chứng thực**

+ *control-flag*: indicate the **continuation** or **failure behaviour** of PAM-API

Type	Ý nghĩa
binding	nếu module successful và **không** có modules nào sau đấy có **flag *required* mà fail**, thì **skip** toàn bộ chỗ module còn lại; nếu failed thì lưu lại *required* failure và tiếp tục xử lý stack
required	nếu module successful, lưu lại *required* success và **tiếp tục check** các module sau đấy; nếu failed (và là cái required fail đầu), lưu lại thông báo lỗi và **tiếp tục check** stack
requisite	nếu module successful, lưu lại *required* success và **tiếp tục check** các module sau đấy; nếu failed, lưu lại *required* failure, trả về thông báo lỗi cho cái *required* failure **đầu tiên**, và **dừng check** các module sau đấy
optional	nếu module successful, lưu lại *optional* success và **tiếp tục check**; nếu failed, lưu lại *optional* failure và **tiếp tục check**
sufficient	nếu module successful và không có modules nào sau đấy có flag *required* mà fail thì **skip** toàn bộ chỗ module còn lại; nếu failed thì lưu lại *optional* failure và **tiếp tục check** stack

(continuation behaviour: xác định nếu có modules đằng sau được check. Tùy thuộc vào response của một module nào đó mà có thể skip thêm bất kỳ module nào

failure behaviour: xác định cách mà tin báo lỗi được report; thường là *optional* hoặc *required*)

+ *module*: đường dẫn tương đối/tuyệt đối của PAM

+ *module-arguments*: list các tokens ngăn cách bởi dấu cách

2. Một số system calls:

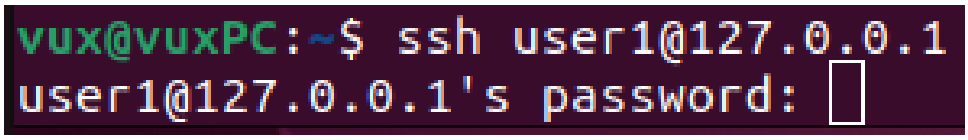
Tên	Công dụng
execve(pathname, argv, envp)	Thực thi chương trình thông qua *pathname*, với tham số *argv* và danh sách biến môi trường *envp*
access(pathname, mode)	Check xem process được gọi có thể truy cập vào *pathname* không
open()	
mmap()	Memory allocation (phân bổ dung lượng)
write()	Output ra terminal

THỰC HÀNH

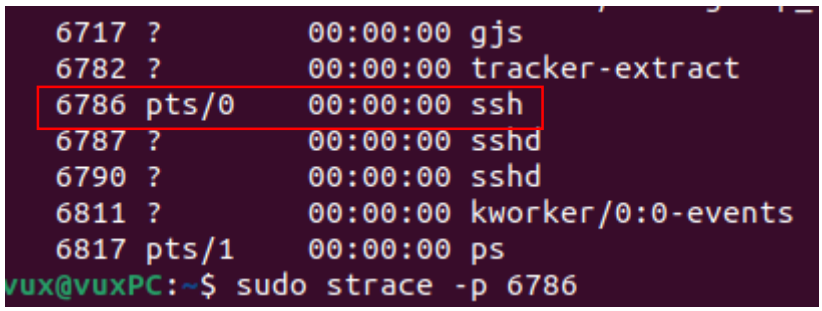
Bài 1: (em chưa làm được ạ :<)

Bài 2:

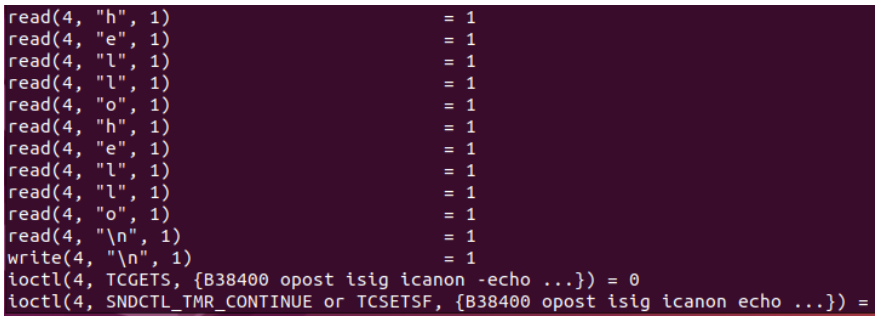
- Trước tiên cần ssh tới máy minh muốn (em dùng tạm localhost vì không hiểu sao em không ssh lên máy khác được ạ :<), nhưng khoản nhập mật khẩu vội



- Mở terminal khác, tìm tiến trình đang chạy lệnh ssh này, sau đó strace tiến trình đó



- Sau đó mới quay lại terminal kia nhập khẩu, khi đó strace sẽ trả lại những system call mà tiến trình gọi. Để ý sau khi nhập xong mật khẩu (và đúng), thì sẽ có những system call sau:



- Những chữ cái viết trong ngoặc kép (trừ \n) sẽ là mật khẩu của user đó

TÀI LIỆU THAM KHẢO:

- SSH:
  + *https://fptcloud.com/ssh-la-gi/* (tiếng Việt)
  + *https://www.techtarget.com/searchsecurity/definition/Secure-Shell* (tiếng Anh)
  + *https://www.hostinger.vn/huong-dan/ssh-la-gi-va-cach-su-dung-ssh-cho-nguoi-moi-bat-dau* (dễ hiểu)
  + *https://www.digitalocean.com/community/tutorials/understanding-the-ssh-encryption-and-connection-process* (tiếng Anh của bài trên, hiểu sâu hơn)
- Dịch vụ SSHD:
  + *https://geekpeach.net/dich-vu-he-dieu-hanh-linux-sshd*
  + *https://www.ssh.com/academy/ssh/sshd*
- OpenSSH:
  + *https://topdev.vn/blog/openssh-la-gi/* (chi tiết)
  + *https://quantrimang.com/gioi-thieu-openssh-131*
- PAM:
  + *https://blogd.net/linux/cau-hinh-su-dung-pam-trong-linux/#:~:text=Linux%20PAM%20l%C3%A0%20g%E1%BB%8Di%20t%E1%BA%AFt,ng%C6%B0%E1%BB%9Di%20d%C3%B9ng%20user%20v%C3%A0%20password*.
  + *https://en.wikipedia.org/wiki/Pluggable_authentication_module*
  + *https://docs.oracle.com/cd/E19683-01/816-4883/pam-36/index.html* (Control Flags)
- strace, ptrace:
  + *https://www.geeksforgeeks.org/strace-command-in-linux-with-examples/* (strace)
  + *https://man7.org/linux/man-pages/man1/strace.1.html* (strace man page)
  + *https://man7.org/linux/man-pages/man2/ptrace.2.html* (ptrace man page)
- Video bổ sung:
  + *https://www.youtube.com/watch?v=ORcvSkgdA58* (giải thích SSH hàn lâm hơn)

+ *https://www.youtube.com/watch?v=NmM9HA2MQGI* và *https://www.youtube.com/watch?v=Yjrfm_oRO0w* và *https://www.youtube.com/watch?v=vsXMMT2CqqE* (giải thích về Diffie-Hellman Key Exchange Algorithm)

+ *https://www.youtube.com/watch?v=GSIDS_lvRv4* (Public-Private Key Encryption)

+ *https://www.youtube.com/watch?v=hQWRp-FdTpc* (Crash course SSH của Traversy, khá recommend)