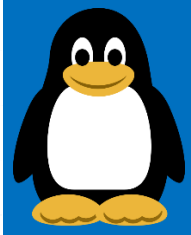

Tài liệu hướng dẫn học Linux LPIC-1

Linux Professional Institute Certification

LPIC-1 102

Giảng viên: Lưu Hồ Phương
Email: phuongluuho@gmail.com

Tiêu đề

Contents

Giảng viên: Lưu Hồ Phương
Email: phuongluuho@gmail.com

Yêu cầu hệ điều hành:

Linux - CentOS 6, CentOS 8 (hoặc bản phân phối SysVinit hoặc Systemd khác)

Chủ đề 105- Shell, Viết kịch bản scripting và Quản lý Dữ liệu

101.1 – Tùy chỉnh và Sử dụng Môi trường Shell (Shell, Biến, Thiết lập và source)

1. Shell đặc biệt:
   - /bin/false - Trả về một mã khác không để chặn bất cứ yêu cầu đăng nhập(login) của người dùng nào
   - /sbin/nologin - Cũng chặn yêu cầu đăng nhập, nhưng trả về một thông báo kết quả dưới dạng văn bản

2 . Biến (môi trường):

- "Lối tắt(shortcut)" đến nhiều giá trị và loại khác nhau
  - **Ví dụ**: BUCKET=123456
    - Việc này sẽ tạo một biến môi trường được gọi là BUCKET và gán giá trị số ban đầu là 123456 vào biến đó
- Có thể hiển thị trên dòng lệnh (hoặc sử dụng như một phần của một lệnh hoặc thay thế) bằng cách đặt ký tự $ trước biến
  - **Ví dụ**: echo $BUCKET
    - Sẽ hiển thị giá trị hiện tại của biến BUCKET (trong ví dụ trước đó, giá trị này sẽ là 123456)
- Biến có thể là số hoặc văn bản và biến văn bản có thể chứa khoảng trắng nếu được đặt trong dấu ngoặc kép hoặc trích dẫn
  - **Ví dụ:** MYVAR= ="Space Value" hoặc MYVAR=Space\ Value
    - Phương pháp đầu tiên (dấu ngoặc kép) là phương pháp phổ biến nhất để gán văn bản cho một biến
- Export
  - Hiển thị các biến đã xuất (khi chạy riêng lẻ)

- Từ khóa, khi đặt trước một biến, cho phép giá trị của biến được truyền cho các shell khác hoặc các tiến trình con của shell hiện tại (hành vi bình thường là giá trị biến chỉ được hiển thị trong shell hiện tại, gọi là phạm vi biến)
- **Ví dụ:**
  - BUCKET=123 * echo $[BUCKET*2]
    - Kết quả sẽ trả về giá trị 246
  - Khi thoát khỏi phiên bash hiện tại và tạo một phiên bash mới, giá trị của biến sẽ không được chuyển sang phiên con mới

3. source (sử dụng các giá trị từ một file khác)
   - Tương đương với việc bao gồm các giá trị được chỉ định trong một file vào một lệnh khác, shell hoặc lệnh khác
   - **Ví dụ:** file sourcefile.sh chứa biến BUCKET=123
     - Lấy giá trị biến từ sourcefile.sh vào shell hiện tại: source ./sourcefile.sh
       - echo $BUCKET từ dấu nhắc lệnh bash sau đó sẽ hiển thị giá trị 123 trong biến
   - Phím tắt cho việc sourcing
     - **Ví dụ**: source ./sourcefile.sh có thể được thay thế bằng ../sourcefile.sh
4. unset
   - Các biến có thể được gỡ bỏ bằng lệnh unset
   - **Ví dụ**: unset BUCKET && echo $BUCKET
     - Trong ví dụ của chúng ta, kết quả sẽ tạo ra đầu ra có giá trị trống vì nó gỡ bỏ biến
5. set
   - Hiển thị tất cả các biến và hàm trong môi trường hiện tại
   - Cũng cho phép kích hoạt/tắt các tính năng shell khác nhau
     - **Ví dụ:** set –x
   - Sẽ in từng lệnh ra terminal khi được thực thi
     - **Ví dụ**: set +x
   - Sẽ tắt (disable)việc in từng lệnh ra terminal

Giảng viên: Lưu Hồ Phương
Email: phuongluuho@gmail.com

105.1 Tùy chỉnh và Sử dụng Môi trường Shell (Môi trường và Bí danh)

1. env
   - Khi chạy lệnh này độc lập(không có tham số gì), hiển thị các biến môi trường hiện tại và giá trị của chúng
   - Có thể được sử dụng để sửa đổi môi trường hiện tại trước khi bạn chạy một lệnh (hoặc một tập lệnh khi được sử dụng ở trên cùng của các dòng lệnh)
     - Ví dụ: #!/bin/env java
       - Dòng lệnh trên sẽ cho phép shell tìm kiếm ứng dụng java trong biến môi trường PATH trước khi chạy các lệnh tiếp theo
   - Thường được sử dụng để xóa môi trường hiện tại trước khi thực hiện một thao tác gì đó
     - -i: Xóa (tạm thời, chỉ dành cho lệnh hoặc tập lệnh) môi trường hiện tại trước khi chạy các lệnh sau đó
     - -c '[lệnh]': Sẽ chạy lệnh được chỉ định, trong một môi trường nguyên bản (khi sử dụng với -i) hoặc môi trường hiện tại (khi không sử dụng với -i)
2. alias
   - Cho phép bạn thay thế một lệnh bằng lệnh khác hoặc đặt một lối tắt bằng một lệnh phức tạp hơn với các chuyển đổi
   - Ví dụ: alias ll='ls -al --color=yes | more'
     - Lối tắt nói chung thường được sử dụng để tạo một bí danh(alias) trong môi trường để hiển thị danh sách full color của thư mục hiện tại, được phân trang bằng more khi cần thiết
   - Có thể được sử dụng để ghi đè lên một lệnh hiện có
     - **Ví dụ:** alias rm=alias rm="echo 'You do not really want to remove that do you?'"
   - Các bí danh có thể bị ghi đè bằng cách:
     - Đặt ký tự thoát trước một lệnh (ví dụ: \rm)
     - Sử dụng đường dẫn đầy đủ đến lệnh (ví dụ: /bin/rm)
3. unalias
   - Xóa (trong môi trường hiện tại) bí danh được chỉ định
   - **Ví dụ**: unalias rm
     - Gỡ bỏ ghi đè lên lệnh rm đã được thiết đặt trong ví dụ trên của chúng ta
   - -a: Xóa TẤT CẢ bí danh được thiết đặt trong môi trường hiện tại trong suốt phiên làm việc

- **Lưu ý**: Cách duy nhất để xóa một bí danh một cách vĩnh viễn là loại bỏ nó khỏi tập lệnh (khởi động hoặc môi trường) đã đặt nó lúc đầu

4. function
   - Cho phép bạn tạo một "ứng dụng nhỏ" có thể thực hiện nhiều công việc
   - Từ khóa function phải xuất hiện ngay ở đầu của định nghĩa.

   Một hàm có thể chạy trên nhiều dòng lệnh và có cấu trúc như sau:

```
function myfunc() {
  cp /path/to/dir /path/to/destination tar cvf destination.tar /path/to/destination
 }
```

   - Khi gọi hàm này (trong trường hợp này được gọi như các lệnh khác trong môi trường shell, bằng cách sử dụng tên của hàm, myfunc), nó sẽ copy các files từ một vị trí tới một vị trí khác và sau đó tạo một file tar từ vị trí đã tạo đó có tên là destination.tar
   - Bạn có thể sử dụng các biến đặc biệt ngoài những gì thông thường có sẵn trong môi trường shell
     - $1: Đại diện cho tùy chọn đầu tiên được truyền vào hàm từ dòng lệnh
     - $*: Phương pháp để tham chiếu đến TẤT CẢ các tùy chọn cùng một lúc
5. .profile
   - Nằm trong thư mục /etc
   - file cấu hình global áp dụng các thiết lập của nó cho tất cả môi trường người dùng (miễn là họ sử dụng bash shell)
   - "Được sourced" (được gọi) mỗi khi đăng nhập (là một tập lệnh kịch bản script)
   - Được sử dụng để đặt các biến môi trường (như PATH hoặc PS1 cho dấu nhắc bash) và một số thiết đặt môi trường có điều kiện khác được dựa trên các điều kiện khác nhau (**ví dụ**: người dùng thường so với người dùng quản trị/root)
6. .bash_profile
   - Nằm trong thư mục home của người dùng (ví dụ: /home/user)
   - "Được sourced" hoặc thực thi theo thứ tự (nếu tồn tại) là một phần của quá trình đăng nhập (sau file global profile)
   - Ảnh hưởng đến môi trường của người dùng hiện tại (và CHỈ môi trường của người dùng đó)
   - **Lưu ý**: Có thể được gọi là .profile - nhưng nếu cả hai tồn tại, chỉ .bash_profile sẽ được áp dụng
7. .bashrc
   - Nằm trong thư mục /etc
   - File cấu hình global áp dụng các thiết lập của nó cho tất cả môi trường người dùng (miễn là họ sử dụng bash shell)
   - "Được sourced" hoặc thực thi sau profile từ .bash_profile của người dùng (ở trên) nếu tồn tại
   - Thông thường được sử dụng để định nghĩa các alias và hàm cho môi trường của tất cả người dùng
8. .bashrc
   - Nằm trong thư mục home của người dùng (ví dụ: /home/user)
   - "Được sourced" hoặc thực thi (nếu tồn tại) từ .bash_profile của người dùng

- Ảnh hưởng đến môi trường của người dùng hiện tại (và CHỈ môi trường của người dùng đó)
- Một file thường khác được sử dụng để định nghĩa các alias, giao diện dấu nhắc bash, hàm cho môi trường, v.v.

9. . bash_login
   - Tương tự như các tập lệnh "profile" khác, nhưng nằm trong thư mục home của người dùng, cho phép bạn đặt các biến môi trường, hàm, alias, v.v. khi đăng nhập
   - **Lưu ý**: Nếu tồn tại .profile hoặc .bash_profile, file này SẼ KHÔNG được đọc/sourced và thực thi
10. .bash_logout
    - Đây là tập lệnh cuối cùng được sourced/thực thi trong một phiên đăng nhập cho người dùng
    - Thông thường được sử dụng để kết thúc các quá trình của người dùng, xóa màn hình từ bất kỳ console nào đã đăng nhập, đăng xuất khỏi bất kỳ ứng dụng nào đang sử dụng thông tin đăng nhập, unmount các chia sẻ cụ thể của người dùng, v.v.

11. Phiên đăng nhập(login session) ví dụ mẫu:

1. Kết nối và đăng nhập bằng tên người dùng/mật khẩu
2. Tập lệnh /home/user/.bash_login được thực thi (giả sử nó tồn tại và .bash_profile hoặc .profile không tồn tại)
3. Tập lệnh /etc/profile được thực thi (giả sử nó tồn tại)
4. Tập lệnh /home/user/.bash_profile được thực thi (giả sử nếu nó tồn tại)
   - Nếu không, thì .profile sẽ được thực thi nếu nó tồn tại
5. Tập lệnh /home/user/.bashrc được thực thi (được sourced từ .bash_profile hoặc .profile của người dùng)
6. Phiên kết thúc với đăng xuất hoặc thoát (hoặc CTL+D), tập lệnh .bash_logout được thực thi

Giảng viên: Lưu Hồ Phương
Email: phuongluuho@gmail.com

105.2 Tùy chỉnh hoặc Viết các Kịch bản(script) Đơn giản (Đọc và Viết file và Thực thi Lệnh)

- Kịch bản Shell
  - Đơn giản là một tập hợp các lệnh thực hiện một số công việc, thường có các tham số tùy chọn được cung cấp khi chạy kịch bản
  - Kịch bản bắt đầu bằng cách xác định loại shell mà chúng sẽ thực thi các lệnh trong đó
    - **Ví dụ**: #!/bin/bash
      - Cho biết rằng kịch bản này sẽ thực thi bằng cách sử dụng lệnh /bin/bash như shell mà nó sẽ chạy trong đó
  - Giải thích(comments) được đánh dấu bằng ký tự #
    - Ví dụ: # This is a comment
      - Tất cả các dòng ký tự sau ký tự # trên sẽ không được biên dịch bởi shell hoặc kịch bản
      - Comments là chỉ có một dòng; nhiều dòng comments phải được tiền đề bằng ký tự # trên mỗi dòng
  - Các lệnh có thể được thực thi trong một kịch bản shell giống như trên dòng lệnh trong Linux bash shell (kịch bản cũng sẽ kế thừa quyền của người thực thi lệnh)
  - Tham số
    - Đây là các chuỗi, số, v.v. được chuyển đến môi trường, cùng với lệnh, mà kịch bản có thể sử dụng
      - **Ví dụ**: ./myscript.sh /home/user/testfile.txt
        - **VD** trên sẽ gọi kịch bản có tên myscript.sh từ thư mục hiện tại và truyền một tham số duy nhất với giá trị /home/user/testfile.txt vào nó
    - Được tham chiếu bằng số thứ tự của tham số
      - $[#]: Số thứ tự của tham số mà bạn sử dụng trong lệnh trong kịch bản của bạn
        - Trong ví dụ trên của chúng ta, nó sẽ chạy lệnh được chỉ định trên file /home/user/testfile.txt được truyền vào khi chúng ta chạy kịch bản
    - Đánh số tham số bắt đầu từ 1, các tham số không tồn tại sẽ không gây lỗi mà chỉ đơn giản được giải thích là "trống"
  - Kịch bản ví dụ: myscript.txt

```
#!/bin/bash
# create an empty file and add some content to it
touch $1
echo "This is a test file" > $1
# clear the screen and then display the file created clear
cat $1
```

  - /usr/bin: Các kịch bản có sẵn trong hệ thống Linux
  - /usr/local/bin: Các kịch bản được cài đặt/tạo ra chỉ dùng cho hệ thống cục bộ của người dùng user
  - /home/user/bin: Các kịch bản được tạo bởi người dùng "user" và chỉ có sẵn cho người dùng đó
  - Quyền thực thi của file kịch bản script

- Hãy đảm bảo bạn thiết lập quyền thực thi cho kịch bản
  - **Ví dụ**: chmod 755 myscript.sh
  - lệnh trên đảm bảo là tất cả người dùng users đều có quyền thực thi chạy kịch bản

105.2 Tùy chỉnh hoặc Viết các Kịch bản(script) Đơn giản (Kiểm tra Chuỗi, Số và file, Lặp(looping) và Câu điều kiện)

1. Câu điều kiện(Conditionals)

- Cho phép bạn kiểm soát các điều kiện mà các lệnh trong kịch bản của bạn sẽ chạy
- if/then/else/fi
  - Cấu trúc của một câu lệnh điều kiện if, theo nghĩa đơn giản: "nếu điều này thì thực hiện điều đó, nếu không thì thực hiện điều khác fi (kết thúc end)"
  - "Fall down" của if/then/else if/else if/else/fi
    - Cấu trúc của một câu lệnh nhiều lệnh và nhiều kiểm tra
  - Được sử dụng để kiểm tra giá trị biến và thực hiện các hành động tương ứng
    - Ví dụ: if [["$?" == "0"]]
      - Sẽ xác định xem lệnh trước đó đã thành công hay không
      - **Lưu ý**: Biến $? là một biến đặc biệt có thể được kiểm tra ngay sau các lệnh. Nếu giá trị là bằng 0, thì lệnh đã thành công (không xảy ra lỗi). Nếu các giá trị khác không, lệnh đã thất bại.
  - Có thể được sử dụng để kiểm tra trực tiếp các lệnh và giá trị
    - Ví dụ: if pgrep sshd
      - Sẽ trả về true nếu tiến trình sshd trả về một PID (tức là đang chạy)
  - !: Thường được sử dụng để kiểm tra xem một cái gì đó KHÔNG PHẢI là một cái gì đó
    - **Ví dụ**: if !pgrep sshd
      - Sẽ trả về true nếu tiến trình sshd KHÔNG đang chạy

2. Kiểm tra(test)
   - Lệnh để "test" các điều kiện khác nhau với một tham số và giá trị
     - **Ví dụ**: test -f /home/user/testfile.txt
       - Sẽ kiểm tra sự tồn tại của một file có tên /home/user/testfile.txt
   - Kết hợp với câu lệnh điều kiện như if, có thể là một phương pháp hiệu quả để kiểm tra các điều kiện khác nhau trong các chuỗi, các số và files

     - -d [file]: file tồn tại và là một thư mục
     - -f [file]: file tồn tại và là một file thường
     - -h [file]: file tồn tại và là một liên kết tượng trưng
     - -r/w/x [file] - file tồn tại và người dùng có quyền đọc, ghi hoặc thực thi (được chỉ định bằng chữ cái)
     - -s [file] - file tồn tại và có kích thước lớn hơn 0 byte
     - **Ví dụ:** if test -d /home/user
       - Sẽ kiểm tra xem file/home/user có phải là thư mục hay không
   - Từ khóa "test" có thể được bỏ hoàn toàn bằng cách sử dụng phương pháp "dấu ngoặc vuông" (tức là [,])

- **Ví dụ**: if [-f /home/user/testfile.txt]
  - Sẽ kiểm tra sự tồn tại của một file có tên /home/user/testfile.txt
  - **Lưu ý**: Cần có một dấu cách sau dấu ngoặc vuông mở và trước dấu ngoặc vuông đóng, nếu không sẽ gặp lỗi "command not found"

- Chuỗi (Strings)
  - Kiểm tra độ dài chuỗi
  - -n [tên chuỗi]: Chuỗi có độ dài khác không
  - -z [tênchuỗi]: Chuỗi có độ dài bằng không
  - Kiểm tra bằng nhau (=) với dấu bằng đơn
    - **Ví dụ:** if [["$MYSTRING" = "Some Value"]]
      - Kiểm tra xem biến $MYSTRING có giá trị "Some Value" không
  - Không bằng nhau (!=)
    - **Ví dụ:** if [["$MYSTRING" != "Some Value"]]
      - Kiểm tra xem biến $MYSTRING có khác "Some Value" không
- Số nguyên (Integers)
  - Toán tử so sánh
  - -eq: Kiểm tra bằng nhau
  - -ne: Kiểm tra không bằng nhau
  - -gt: Lớn hơn
  - -ge: Lớn hơn hoặc bằng
  - -lt: Nhỏ hơn
  - -le: Nhỏ hơn hoặc bằng
    - Ví dụ: if [[$MYAGE -gt 50]]
      - Sẽ kiểm tra xem biến $MYAGE có lớn hơn 50 không

- Multiple tests (Kiểm tra nhiều điều kiện)
  - Có thể kết hợp bằng toán tử && (và) hoặc toán tử || (hoặc)
    - **Ví dụ:** if [[$MYAGE -gt 50 && $MYAGE -lt 100]]
      - Sẽ kiểm tra xem biến $MYAGE có lớn hơn 50 nhưng nhỏ hơn 100 không (việc này có nghĩa là tôi già nhưng có thể chưa chết ☺)

3. Looping (Vòng lặp)
   - Cho phép xác định số lần một phần cụ thể trong script của bạn sẽ chạy
   - for/do/done
     - Lặp qua một số lượng cố định các mục
       - Ví dụ: for city in Hanoi Haiphong Saigon; do
         - Sẽ thực thi các lệnh sau đó qua ba thành phố trong danh sách
   - while/done
     - Lặp lại cho đến khi điều kiện được chỉ định là sai, lặp lại trong khi điều kiện đúng
       - Ví dụ: while [[$MYLOOP -lt 10]]
         - Sẽ lặp lại miễn là biến $MYLOOP nhỏ hơn 10
   - until
     - Lặp lại giống như while nhưng có dấu ! ở cuối (tiếp tục khi sai, dừng khi đúng)
   - seq
     - Hữu ích khi bạn cần lặp lại một số lần đã biết
     - -w: Thêm các số không hàng đầu vào đầu ra khi độ dài số thay đổi
     - [#] [#] [#] - số bắt đầu, số đếm, số kết thúc (**Lưu ý**: Khi chỉ cung cấp hai tham số, chúng là số bắt đầu/số kết thúc)

4. read
   - Hữu ích để đọc đầu vào từ một console (và/hoặc file); tương tự như lệnh cat
   - Có thể được sử dụng để nhận đầu vào từ người dùng
     - **Ví dụ**: read FIRSTNAME
       - Sẽ yêu cầu người dùng nhập FIRSTNAME và lưu nó trong biến đó để sử dụng sau này
   - exit
     - Cho phép bạn trả về một mã lỗi không bằng không đến môi trường, sau đó có thể được read/giám sát/phản ứng bởi các ứng dụng khác
       - Ví dụ: exit 66
         - Sẽ trả về mã lỗi "66" cho môi trường và lệnh echo $? sau đó sẽ trả về giá trị đó

Chủ đề 106 - Giao diện người dùng và Desktop

106.1 Cài đặt và Cấu hình X11 (Cài đặt Hệ thống XWindow, Desktop và file Cấu hình Xorg)

1. X11 / Xorg
   - X11 đã là giao diện người dùng đồ họa từ đầu cho Unix và Linux
   - Xorg, mặc dù hoàn toàn tương thích, được thiết kế để thay thế nó và cung cấp sự trừu tượng bổ sung về bảo mật và cấu hình hiệu suất
2. Trình quản lý cửa sổ (Window managers)
   - Thêm các thành phần cho "cửa sổ" được vẽ lên màn hình bởi máy chủ X (thanh cuộn, nút max/min, v.v.)
3. Desktops
   - Phần tương tác của GUI; đặt kiểu, biểu tượng, desktop ảo, v.v.
   - **Ví dụ:**
     - KDE
     - Gnome
     - Unity
     - XFCE
   - Các thành phần thông thường trong desktop:
     - Media players
     - Trình quản lý cửa sổ
     - Control panel (settings)
     - Themes/styles
     - File manager
4. Cài đặt
   - Cách cài đặt sẽ khác nhau trên hầu hết các bản phân phối Linux
   - Ví dụ: yum groupinstall "X Window System" "Desktop" "Desktop Platform"
   - Ví dụ: sudo apt-get install xserver-xorg-core unity
   - Sẽ cài đặt tất cả các thành phần cần thiết cho Hệ thống XWindow nói chung và sau đó thêm môi trường desktop mặc định cho một bản phân phối CentOS 6 nếu nó chưa có (**ví dụ**: cài đặt server hoặc cài đặt tối giản nhất)
5. /etc/X11/xorg.conf
   - file cấu hình chính cho Hệ thống X Window

- Các phần chính (ví dụ)
  - Files: Các files được sử dụng bởi máy chủ X (**ví dụ**: font chữ)
  - Module: Thiết bị(devices)
  - InputDevice: Bàn phím và chuột (và có thể là bàn phím đặc biệt, nếu được phát hiện)
  - Device: Card đồ họa và bất kỳ tham chiếu trình điều khiển nào
  - Monitor: Các màn hình được phát hiện
  - Screen: Mô tả các độ phân giải và độ sâu màu được hỗ trợ cho việc kiểm tra phát hiện màn hình và X server

- Có thể được tạo ra (giá trị phát hiện cơ bản)
  - X -configure
  - Thông thường sẽ tạo ra một file có tên là xorg.conf.new trong thư mục gốc của người dùng đang chạy lệnh này
  - Xorg -configure
    - Nếu lệnh trên không hoạt động, các phiên bản X mới sẽ sử dụng lệnh này

6. XFS
   - font server cho X
   - Cung cấp quyền truy cập vào các font chữ cho X server
   - Kiểm tra:
     - service xfs status
       - Sẽ xác định xem nó có đang chạy trên hệ thống Red Hat/CentOS hay không
7. FontPath
   - Nếu máy chủ font chữ không được sử dụng, có thể tồn tại trong file xorg.conf
     - **Ví dụ**: FontPath "/usr/X11R6/lib/X11/fonts/100dpi
       - Đặt đường dẫn để tìm kiếm các font chữ 100dpi được sử dụng bởi X server.

106.1 Cài đặt và cấu hình X11 (X Utilities cho Screen và Window Information)

1. xwininfo
   - Hiển thị thông tin về cửa sổ được chọn trên màn hình desktop
   - Chạy lệnh này sẽ hiển thị một con trỏ đặc biệt để chọn cửa sổ cần truy vấn thông tin
2. xdpyinfo
   - Cung cấp thông tin về phiên X hiện tại
   - **Đề tài kiểm tra**: Các câu hỏi thông thường liên quan đến các tiện ích có thể sử dụng để hiển thị độ phân giải màn hình và độ sâu màu
3. xhost
   - Kiểm soát quyền truy cập vào chính X server
   - Khi chạy một mình, nó hiển thị trạng thái truy cập hiện tại
   - + [IP]: Vô hiệu hóa kiểm soát truy cập, mở máy chủ cho phép kết nối từ mọi máy chủ (hoặc chỉ từ IP đã chỉ định)
   - -: Bật kiểm soát truy cập, chỉ cho phép các clients được phân quyền được phép kết nối

4. DISPLAY: Biến DISPLAY trong X11 là một biến môi trường được sử dụng để chỉ định màn hình hiển thị X mà các ứng dụng X sẽ sử dụng.

106.2 Thiết lập một Trình quản lý Hiển thị

1. xdm
   - Trình quản lý hiển thị là một phần của gói phần mềm Xorg
   - /usr/bin
     - Là thư mục nơi nó tồn tại trong đó, nếu đã được cài đặt
   - xorg-x11-xdm
     - Gói cho trình quản lý hiển thị
   - Thông thường không được cài đặt/sử dụng, trừ khi không sử dụng môi trường desktop đầy đủ
   - /etc/X11/xdm
     - Thư mục cấu hình
2. kdm
   - Trình quản lý hiển thị KDE (cũ)
   - Đã được thay thế bằng kwin và sau đó là lightdm (xem bên dưới)
   - /etc/kde/kdm
     - Thư mục cấu hình
   - /usr/bin
     - Đường dẫn chứa file thực thi
3. gdm
   - Trình quản lý hiển thị Gnome
   - /etc/gdb
     - Thư mục cấu hình
   - /usr/bin
     - Thư mục chứa filethực thi
4. lightdm
   - Hoạt động như một dịch vụ
     - systemctl status lightdm (hệ thống systemd)
   - /etc/lightdm/lightdm.conf
     - Nếu tồn tại, chứa cấu hình cho trình quản lý hiển thị
   - Thiết kế để là một trình quản lý hiển thị nhẹ hơn (thay thế kdm)
   - /usr/share/doc/lightdm
     - Thư mục chứa file cấu hình mẫu
5. Chức năng chung cho tất cả các trình quản lý hiển thị
   - Xử lý đăng nhập(login) vào môi trường desktop
6. Thay đổi trình quản lý desktop ưa thích
   - Red Hat/CentOS (Phiên bản cũ v6 và trước đó)
     - /etc/X11/prefdm: Kịch bản kiểm tra trình quản lý hiển thị ưu tiên
       - Đặt giá trị $DISPLAYMANAGER thành "lightdm" (không có sẵn, ngoại trừ phiên bản mới nhất của CentOS)

- Debian/Ubuntu
  - dpkg-reconfigure [trình quản lý hiện tại]: Cho phép cấu hình lại trình quản lý hiển thị, màn hình sẽ yêu cầu thông tin cho bất kỳ DM đã được cài đặt

106.3 Accessibility

1. Sticky/repeat keys (các phím bấm dính/lặp lại)
2. Slow/bounce/toggle keys (Các phím chậm/nhảy/toggle)
3. Mouse keys (Phím chuột)
4. High contrast/large print (Chế độ tương phản cao/in ấn)
5. Screen reader
   - Cài đặt trên một số desktop
   - Orca
   - emacspeak (cho trình soạn thảo emacs)
6. Hiển thị Braille
7. Phóng to màn hình
8. Bàn phím ảo trên màn hình
   - GOK (Gnome On-screen Keyboard)

Chủ đề 107 - Các nhiệm vụ quản trị

107.1 Quản lý tài khoản người dùng(user) và nhóm(group) và các files hệ thống liên quan (UIDs và GIDs, các files mật khẩu)

1. UID
   - User ID (ID người dùng)
   - Là một số ID từ 0 đến hơn 4 tỷ
   - Tài khoản người dùng thường sẽ nằm trong khoảng từ 500 đến 65,000
2. GID
   - Group ID (ID nhóm)
   - Một số ID từ 0 đến hơn 4 tỷ
3. Các UID đặc biệt
   - UID 0: Người dùng root/quản trị hệ thống
   - UID 1: Người dùng bin (các files nhị phân hệ thống Linux và tài khoản không đăng nhập)
   - UID 48: Người dùng Apache (nếu đã cài đặt)
   - UID 99: Tài khoản nobody (được sử dụng cho nhiều mục đích, ví dụ như truy cập

không đăng nhập FTP, cũng có thể ánh xạ đến một tài khoản root cho một số cấu hình NFS cụ thể (**ví dụ** như tùy chọn root_squash))

4. Các GID đặc biệt
   - GID 0: Nhóm root/quản trị (các thành viên trong nhóm này có quyền truy cập vào các tài nguyên hạn chế có hạn chế quyền truy cập)
   - GID 1: Nhóm bin (các file nhị phân hệ thống và tài khoản không đăng nhập)
   - GID 100: Nhóm người dùng users group (đặt người dùng vào nhóm này để cung cấp quyền truy cập vào các tài nguyên bằng cách gán quyền sở hữu nhóm cho nhóm này)
5. /etc/passwd
   - Định nghĩa thông tin cụ thể cho người dùng như thư mục home, ánh xạ tên người dùng thành UID, shell đăng nhập mặc định, v.v.
   - Quyền mặc định:
     - Red Hat -rw-r--r-- (644)
     - Debian -rw-r--r-- (644)
   - Các ID được định nghĩa ở đây nằm trong khoảng từ 0 đến 499 (các tài khoản hệ thống được định nghĩa trước; các tài khoản người dùng nằm trong khoảng đó)
   - **Ví dụ**: user:x:1001:1001::/home/user:/bin/bash
     - [tên người dùng]:[mật khẩu trong file shadow]:[UID]:[GID(primary)]:[Mô tả]:[thư mục home]:[shell đăng nhập mặc định]
6. Các shell đặc biệt
   - /bin/false: Người dùng không thể đăng nhập (được sử dụng cho các tài khoản hệ thống)
   - /etc/nologin: Hiển thị thông báo là tài khoản không khả dụng (nếu /etc/nologin.txt tồn tại, sẽ hiển thị thông báo của nội dung trong file này)
7. /etc/shadow
   - Ánh xạ các tên người dùng và (nếu tài khoản có mật khẩu đã gán) giá trị mật khẩu băm của họ
     - Quyền mặc định:
       - Red Hat -r------- (400)
       - Debian rw-r----- (640)
   - **Lưu ý**: Xóa mật khẩu (giá trị băm) từ đây có thể cho phép bạn truy cập vào tài khoản người dùng đó nếu bạn có quyền sudo
     - **Ví dụ**: user:6ejcjmixY$AvmZqcTyRcdmbkdK0BOOYZzwM7Q0g /XnWsvK2Ky1Hxd.EkiL6NA4daiH
     - [tên người dùng]:[mật khẩu được mã hóa]:[lần thay đổi mật khẩu cuối cùng]:[số ngày tối đa trước khi yêu cầu thay đổi mật khẩu]:[số ngày cảnh báo trước khi hết hạn mật khẩu]:[khoảng thời gian gia hạn sau khi hết hạn]:[ngày hết hạn]:[dành riêng]
     - Tất cả điều này được điều khiển trên các hệ thống Linux cũ bằng Bộ công cụ Shadow, các hệ thống mới sử dụng các mô-đun PAM mạnh mẽ hơn.
8. /etc/group
   - Định nghĩa thông tin cụ thể cho nhóm(group) như ánh xạ tên nhóm thành GID, các thành viên khác trong nhóm, v.v.
   - Nhóm có thể chứa nhiều người dùng.

     - **Ví dụ**: user:x:1001:test,temp • [tên nhóm]:[mật khẩu nhóm trong file shadow]:[GID]:[các thành viên nhóm phụ - nếu có]

9. **Lưu ý**: Khi đăng nhập vào hệ thống Linux, người dùng PHẢI sử dụng tên người dùng của họ. Đăng nhập qua UID không được phép, ngay cả khi UID tồn tại.
10. getent
    - Tiện ích cho phép bạn tìm kiếm thông tin tài khoản từ các nguồn cục bộ (/etc/passwd và /etc/shadow) và các nguồn network (ví dụ: LDAP).

    - **Ví dụ**: getent passwd user

11. /etc/nsswitch.conf

    - Xác định thứ tự tìm kiếm thông tin tài khoản người dùng trên hệ thống trong quá trình đăng nhập
      - **Ví dụ**:
        password: files nis
        shadow: files nis
        - Sẽ tìm kiếm thông tin tài khoản trong các files LOCAL và sau đó tìm kiếm trong cơ sở dữ liệu người dùng từ xa được cấu hình nếu không tìm thấy.
12. Các files đăng nhập đặc biệt
    - /etc/motd: Nếu tồn tại, hiển thị nội dung của file này
    - .hushlogin: Nếu tồn tại trong thư mục home của người dùng, ngăn đăng nhập kiểm tra thư và hiển thị thông tin đăng nhập trước đó.
    - /etc/login.defs: Thiết lập mặc định cho người dùng khi tạo bằng lệnh useradd.
    - /etc/securetty: Xác định nơi mà người dùng root được phép đăng nhập (nếu không tồn tại, root có thể đăng nhập từ bất cứ đâu).
    - /etc/usertty: Xác định các tham số cho đăng nhập người dùng (vị trí, ngày hoặc giờ, v.v.)
      - **Lưu ý:** Chỉ được sử dụng nếu hệ thống không có các mô-đun PAM (mô-đun xác thực có thể gắn kết).

107.1 Quản lý tài khoản người dùng và nhóm và các files hệ thống liên quan (Quản lý tài khoản người dùng)

1. **LƯU Ý**: Mặc dù có thể tạo người dùng bằng cách chỉnh sửa trực tiếp các files /etc/passwd và /etc/group, việc này có thể đưa đến các rủi ro về bảo mật VÀ có thể chứa lỗi gây ngăn chặn đăng nhập người dùng hoặc gây ra hậu quả không mong muốn.
2. 2. useradd
3. • Tiện ích tiêu chuẩn để thêm người dùng mới vào hệ thống Linux
4. • Có thể hoạt động khác nhau trên các hệ thống dựa trên Red Hat và dựa trên Debian, tốt nhất là chạy lệnh với các tham số rõ ràng
5. • -m: Tạo thư mục home
   • -d [thư mục]: Chỉ định để tạo thư mục home
   • -k [thư mục mẫu]: Sao chép nội dung của thư mục được chỉ định (thông thường là /etc/skel) vào thư mục home của người dùng mới
   • -g [GID chính]: Chỉ định GID chính của người dùng
   • -u [UID]: Gán UID chỉ định cho người dùng (**Lưu ý:** Sẽ báo lỗi nếu UID đã tồn tại)

- -e [ngày]: Ngày sau đó tài khoản này sẽ bị vô hiệu hóa
- -G [GID phụ]: Cho phép bạn đặt một nhóm phụ cho người dùng

- -f [# ngày]: đặt số ngày sau khi mật khẩu đạt tới thời hạn đăng nhập tối đa mà tài khoản vẫn cho phép đăng nhập

- -o [UID không duy nhất]: Cho phép tạo một người dùng với UID không duy nhất (trùng lặp)

- -s [đường dẫn và tên file của shell đăng nhập]: Đường dẫn đầy đủ và tên shell đăng nhập mặc định cho người dùng

3. /etc/skel
   - Nội dung của thư mục này có thể được sao chép vào thư mục home của người dùng mới tùy thuộc vào cách người dùng được thêm vào

4. /etc/default/useradd
   - File Chứa các giá trị mặc định cho lệnh useradd khi các tham số đó không được sử dụng
5. groupadd
   - Giống như useradd, sẽ thêm một nhóm mới như đã chỉ định
   - **Lưu ý**: Không thêm người dùng vào nhóm
   - -g [GID]: Tạo nhóm với GID được chỉ định
6. usermod
   - Sửa đổi các đặc điểm và/hoặc thành viên của người dùng hiện có
   - -c [mô tả]: Sửa đổi mô tả người dùng trong file /etc/passwd
   - -d [thư mục home mới]: Thay đổi thư mục home của người dùng
   - -e [ngày]: Thay đổi ngày hết hạn tài khoản
   - -f [# ngày]: Thay đổi số ngày sau khi mật khẩu đạt tới tuổi tối đa mà tài khoản vẫn cho phép đăng nhập
   - -g [GID]: Thay đổi GID chính của người dùng
   - -G [GID]: Thay đổi GID phụ(s) của người dùng (có thể là nhiều nhóm trong danh sách phân cách bằng dấu phẩy)
   - -s [đường dẫn và tên tệp của shell đăng nhập]: Thay đổi đường dẫn đầy đủ và tên shell đăng nhập mặc định cho người dùng
   - -u [UID]: Thay đổi UID (**Lưu ý**: Sẽ thay đổi thư mục home để phù hợp, nhưng không thay đổi các file sở hữu bởi người dùng khác)
   - -L: Khóa tài khoản người dùng
   - -U: Mở khóa tài khoản người dùng
7. groupmod
   - Sửa đổi các đặc điểm của nhóm được chỉ định
   - -g [GID]: Thay đổi GID của nhóm được chỉ định
8. userdel
   - Xóa tài khoản người dùng được chỉ định
   - -r: Cũng xóa tất cả mails của người dùng, print jobs của người dùng, các cron jobs và toàn bộ nội dung thư mục home của người dùng (**Lưu ý**: Tất cả các files khác

thuộc sở hữu của người dùng vẫn tồn tại, sở hữu sẽ chuyển về UID của người dùng đã bị xóa, trở thành files mồ côi)
- **Lưu ý**: Bạn không thể xóa một người dùng đang được sử dụng hoặc có một tiến trình liên quan đến tài khoản đó

9. groupdel
- Xóa nhóm được chỉ định
- Các files/thư mục thuộc sở hữu của nhóm sẽ chuyển về GID; bạn có thể thay đổi sở hữu tại thời điểm đó

10. passwd
- Đặt/thay đổi mật khẩu của người dùng được chỉ định
- **Lưu ý**: Người dùng có thể thay đổi mật khẩu của chính họ, nhưng không thể thay đổi mật khẩu của người dùng khác (mà không cần sudo), root có thể thay đổi mật khẩu của tất cả các người dùng trong Linux

11. chage
- Thay đổi các tham số về thời hạn tài khoản và mật khẩu của người dùng được chỉ định
- Thay đổi các giá trị trong file /etc/shadow
- -m [# ngày]: Thời gian người dùng phải chờ (tính bằng ngày) giữa các lần thay đổi mật khẩu
- -M [# ngày]: Thời gian trước khi người dùng phải thay đổi mật khẩu
- -d [ngày]: Đặt giá trị đã thay đổi cuối cùng cho mật khẩu
- -E [ngày]: Thay đổi giá trị ngày hết hạn tài khoản
- -I [# ngày]: Số ngày không hoạt động sau khi hết hạn hoặc vượt quá giới hạn tối đa trước khi tài khoản bị khóa
- -W [# ngày]: Cảnh báo về số ngày trước khi người dùng phải thay đổi mật khẩu
- -l: Hiển thị tất cả các giá trị cho người dùng được chỉ định

107.2 Tự động hóa các nhiệm vụ quản trị hệ thống bằng cách lập lịch công việc (cron, crond và crontab)

1. cron
- Là hệ thống lập lịch chính cho các công việc trong Linux
- Công việc được cấu hình để chạy theo lịch trình cố định (một lần hoặc nhiều lần tùy theo yêu cầu)
- crond
  - Dịch vụ chịu trách nhiệm đảm bảo là các công việc cron được chạy
- /etc/cron.*
  - cron.d: Thư mục cấu hình lịch trình công việc tùy chỉnh (công việc cron hệ thống)
  - cron.hourly: Các công việc chạy hàng giờ
  - cron.daily: Các công việc chạy hàng ngày
  - cron.weekly: Các công việc chạy hàng tuần

- cron.monthly: Các công việc chạy hàng tháng

- **Lưu ý**: Trong các thư mục ngoại trừ cron.d, các thư mục này chỉ chứa các tập lệnh mà không bao gồm thông tin lập lịch khác và chúng sẽ không chạy vào cùng thời điểm, nhưng sẽ chạy trong khoảng thời gian được chỉ định.

2 .crontab

- Tiện ích cho phép tạo công việc (định nghĩa lập lịch riêng cho người dùng chạy lệnh)
  - -l: Liệt kê tất cả các công việc cron cho người dùng đã đăng nhập
  - -e: Chỉnh sửa các công việc cron cho người dùng đã đăng nhập
  - -u [tên người dùng]: Áp dụng tùy chọn cho người dùng được chỉ định

3. Định dạng một mục công việc cron
   - [Phút 0-59] [Giờ 0-23] [Ngày trong tháng 1-31] [Tháng 1-12] [Ngày trong tuần 0-7] [CMD]
     - **Lưu ý**: Nếu là công việc cron hệ thống Linux nằm trong /etc/cron.d, sẽ có một trường thứ sáu chỉ định người DÙNG nào sẽ chạy cron đó (xem **ví dụ** /etc/cron.d/raid-check)
     - **Lưu ý**: Trong trường Ngày trong tuần, số 0 và 7 đều chỉ là Chủ nhật
     - **Lưu ý**: Với giờ, 0 là nửa đêm
   - Mỗi cột phải có một giá trị, ngay cả khi giá trị là * có nghĩa là áp dụng cho tất cả các giá trị có thể trong trường
     - **Ví dụ**: 30 23 * 2 7
       - Công việc này sẽ được lên lịch vào 30 phút sau 23 giờ trong ngày (11:30 PM), vào bất kỳ ngày nào của tháng 2 cũng là ngày Chủ nhật
       - **Lưu ý**: Bằng cách sử dụng viết tắt ba chữ cái, bạn có thể thay thế tên của tháng cho trường thứ tư cũng như tên của ngày cho trường thứ năm
       - **Ví dụ**: 30 23 * feb sun
         - Tương đương với ví dụ trước đó
     - Khoảng giá trị
       - Một khoảng giá trị có thể được xử lý bằng cách sử dụng dấu gạch ngang giữa hai giá trị
         - **Ví dụ**: 30 23 * 2-4 7
           - Công việc này sẽ được lên lịch vào 30 phút sau 23 giờ trong ngày (11:30 PM), vào bất kỳ ngày nào của các tháng từ tháng 2 đến tháng 4, cũng là ngày Chủ nhật
       - Cũng có thể được chỉ định bằng danh sách phân cách bằng dấu phẩy
         - **Ví dụ**: 30 23 * 2,3,12 1,3
           - Công việc này sẽ được lên lịch vào 30 phút sau 23 giờ trong ngày (11:30 PM), vào bất kỳ ngày nào của các tháng tháng 2, tháng 3 và tháng 12, cũng là thứ Hai hoặc thứ Tư
       - Giá trị cũng có thể được chỉ định dưới dạng "bước"

- **Ví dụ**: 0 */4 * 2,3,12 1,3
  - Công việc này sẽ được lên lịch vào đầu giờ, bốn giờ một lần, vào bất kỳ ngày nào của các tháng tháng 2, tháng 3 và tháng 12, cũng là thứ Hai hoặc thứ Tư

- Công việc / lệnh để chạy
  - Mọi lệnh shell hợp lệ, ứng dụng hoặc kịch bản
    - **Lưu ý**: Công việc cron cho một người dùng sẽ có môi trường rất giới hạn (.bash_profile, .bashrc, .bash_login, vv.) không chạy, vì vậy cần cung cấp đầy đủ đường dẫn đến các lệnh và biến trong kịch bản, nếu không bạn có thể gặp lỗi hoặc kết quả không mong muốn (**ví dụ**: PATH chỉ có /usr/bin và /bin)
    - **Lưu ý**: Bạn có thể đặt một số biến môi trường đặc biệt trước công việc của bạn
    - **MAILTO**: Đầu ra từ công việc sẽ được gửi qua email đến địa chỉ được chỉ định
    - **SHELL**: Chạy công việc với shell được chỉ định
    - **CRON_TZ**: Sử dụng múi giờ được chỉ định cho crontab
- Chuyển hướng công việc
  - Ngoại trừ biến MAILTO đã được đề cập, bạn có thể chuyển hướng đầu ra vào /dev/null hoặc một file khác
    - **Ví dụ**: 0 */4 * 2,3,12 1,3 /path/application/script.sh > /dev/null
      - Sẽ "loại bỏ" toàn bộ đầu ra (/dev/null)
    - **Ví dụ**: 0 */4 * 2,3,12 1,3 /path/application/script.sh 2>&1 > /root/some.log
      - Sẽ lấy toàn bộ đầu ra chuẩn và lỗi và chuyển hướng vào file /root/some.log

4. /var/spool/cron
   - Tất cả các crontab của người dùng được tạo / chỉnh sửa bằng crontab được lưu tại đây

5./etc/cron.allow
   - Whitelist (Danh sách trắng) của các người dùng được phép chạy các công việc cron
   - Nếu file này tồn tại và trống, thì chỉ root mới có thể truy cập vào crontabs

6. /etc/cron.deny
   - Blacklist(Danh sách đen) chứa tên người dùng users không được phép chạy các công việc cron
   - Nếu file này tồn tại và trống, tất cả các người dùng có thể truy cập vào crontabs và chạy công việc
   - **Lưu ý:** Thứ tự ưu tiên sẽ áp dụng cron.allow và bỏ qua cron.deny nếu nó tồn tại

107.2 Tự động hóa Các tác vụ Quản trị Hệ thống bằng cách Lập lịch Công việc (Anacron và các Lệnh được Lập lịch khác)

1. anacron
   - "Simplified" cron, được sử dụng để bổ sung cho crond
   - Chạy các công việc có thể chạy với độ chính xác thời gian thấp hơn, đặc biệt là đáp ứng các công việc đã được lên lịch khi hệ thống Linux đã tắt
2. /etc/anacrontab
   - Đây là nơi định nghĩa tất cả các công việc
   - Tất cả các công việc sẽ chạy dưới tài khoản root
   - Không có công việc nào được chạy nhiều hơn một lần trong một ngày
   - Tất cả các công việc được chạy liên tiếp
   - Dịch vụ này có sẵn, nhưng thường được chạy từ cron.daily
3. Định dạng của công việc
   - Biến môi trường ở đầu
   - [kỳ hạn trong ngày] [độ trễ trong phút] [định danh công việc] [lệnh chạy]
     - **Ví dụ**: 1 5 cron.daily nice run-parts /etc/cron.daily
       - Chạy các tập lệnh cron.daily một lần mỗi ngày, bị trễ 5 phút tính từ khi hệ thống Linux khởi động, chạy tất cả các công việc đó bằng cách sử dụng tiện ích nice để chạy từng file trong thư mục được chỉ định, trong trường hợp này là /etc/cron.daily
4. Các công việc được lập lịch theo lô
   - Đôi khi được gọi là "ad hoc"; bạn chỉ muốn chạy một cái gì đó trong tương lai, nhưng chỉ một lần
5. at
   - Có thể chưa được cài đặt trong Linux, cài đặt gói at
   - Khởi động dịch vụ atd
   - Phương pháp để chạy các công việc ad hoc
   - Sử dụng lệnh đặc biệt để nhập lệnh chạy vào thời gian được lập lịch
     - **Ví dụ**:

       at 11pm today
       at> wall "Log off everyone" <-- kết thúc với CTL+D
       at> <EOT>

- Sẽ lập lịch cho lệnh wall hiển thị thông báo được chỉ định vào lúc 11 giờ tối hôm nay
- Ví dụ: echo "wall "Log off everyone" " | at 23:00 today
- Phương pháp thay thế cho công việc tương tự
- -l: Liệt kê tất cả các công việc của at hiện đang được lập lịch

6. atq
   - Hiển thị tóm tắt của tất cả các công việc đã lập lịch bằng at
   - Sẽ không hiển thị chi tiết, mà chỉ hiển thị thời gian và ID công việc
7. atrm
   - Cho phép bạn xóa công việc có ID được chỉ định
8. /var/spool/at
   - Thư mục chứa các công việc cần chạy
9. /etc/at.allow
   - Danh sách trắng (whitelist) chứa users các người dùng được phép chạy công việc at
   - Nếu file này tồn tại và trống, thì chỉ có root mới có quyền truy cập vào at
10. /etc/at.deny
    - Danh sách đen (blacklist), chứa users các người dùng không được phép chạy công việc at
    - Nếu file này tồn tại và trống, tất cả người dùng đều có thể truy cập vào at và chạy công việc
    - **Lưu ý**: Thứ tự ưu tiên sẽ áp dụng at.allow và bỏ qua at.deny nếu file này tồn tại

107.3 Địa phương hóa(Localization) và quốc tế hóa(Internationalization) (Ngày tháng-Dates và múi giờ-Timezones)

1. Xác định thời gian trên hệ thống Unix/Linux
   - Thời gian được theo dõi từ "epoch", là một ngày đặc biệt được biết đến là "ngày sinh nhật" của các hệ thống Unix, tức là ngày 1 tháng 1 năm 1970
   - Nó tồn tại dưới dạng một dấu thời gian, tính bằng giây, kể từ thời điểm đó

2. date
   - Hiển thị ngày và giờ hiện tại cùng với múi giờ được cấu hình
   - Bạn có thể kiểm soát định dạng đầu ra bằng ký tự + theo sau là các chỉ thị định dạng đặc biệt
   - %d: Ngày hai chữ số
   - %H: Giờ hai chữ số, định dạng 24 giờ
   - %m: Tháng hai chữ số
   - %M: Phút hai chữ số
   - %Y: Năm bốn chữ số
   - %z: Độ lệch múi giờ
   - Ví dụ: date +"%H:%M-%m%d%Y-%z" •
     Sẽ hiển thị thời gian như "23:11-04212022-0500"
3. Múi giờ

- Một tên đại diện cho "độ lệch" so với UTC (Universal Coordinated Time), còn được gọi là GMT (Greenwich Mean Time)

4. /usr/share/zoneinfo
   - Thư mục cấp cao chứa tất cả các định nghĩa múi giờ
   - Lưu ý: Đây là các file nhị phân và không thể đơn giản chỉ xem trên giao diện console
5. /etc/localtime
   - Múi giờ hệ thống (có thể là bản sao đầy đủ của múi giờ hoặc một liên kết đến múi giờ được cấu hình)
6. TZ (biến)
   - Cho phép bạn ghi đè cài đặt múi giờ của hệ thống trên thư mục trên
   - Thường được đặt trong thư mục người dùng cá nhân như một phần của file .bashrc
7. tzselect
   - Cho phép bạn tìm tên của múi giờ mà bạn muốn sử dụng
   - **Lưu ý**: Lệnh này không thay đổi múi giờ
8. timedatectl (hệ thống dựa trên Red Hat) - dpkg-reconfigure tzdata (hệ thống dựa trên Debian)
   - list-timezones: Liệt kê tất cả các múi giờ để lựa chọn
   - set-timezone [quốc gia/múi giờ]: Đặt thành múi giờ được chỉ định
   - Thực hiện các thay đổi thực tế, đặt cài đặt hệ thống /etc/localtime thành múi giờ đã chọn
9. /etc/timezone
   - File cục bộ được các hệ thống Debian sử dụng để lưu trữ tên của múi giờ được cấu hình

107.3 Địa phương hóa(Localization) và quốc tế hóa(Internationalization (Ngôn ngữ-locales và Mã hóa ký tự-Encoding)

1. locale
   - Cách biểu diễn ngôn ngữ, quốc gia và loại mã hóa của bạn
   - -a: Hiển thị các locales được cài đặt trong hệ thống Linux của bạn
2. Linux và locales
   - GNU gettext
   - Thư viện xử lý quốc tế hóa
   - Phụ thuộc vào biến môi trường để xác định ngôn ngữ cần thiết
     - LANGUAGE: Sử dụng khi hiển thị thông điệp, nhưng không ảnh hưởng đến định dạng
     - LC_ALL: Bắt buộc sử dụng locale chỉ định ngay cả khi các biến khác được đặt khác nhau
     - LC_XXX: Biến quản trị để ghi đè locale cho bất kỳ phần tử nào trong locale
     - LANG: Loại mã hóa
3. ASCII
   - Mã hóa thông tin trao đổi tiêu chuẩn Mỹ

- Được mã hóa thành 7 bit, tạo ra tổng cộng 128 ký tự có thể có.
- Mã hóa tiếng Anh

- Khi các ký tự đó đã được sử dụng hết, việc lưu trữ được thực hiện với 8 bit, tạo ra thêm 128 ký tự có thể có

4. Trang mã
   - Định nghĩa tập ký tự (nơi các ký tự không sử dụng được gán cho ký tự riêng)
   - Đặc thù của mỗi nhà cung cấp, việc thay đổi giữa các ký tự yêu cầu thay đổi trang mã
   - Tiêu chuẩn ISO-8859
   - Định nghĩa các trang mã tiêu chuẩn
     - ISO-8859-1: Bảng chữ cái Latin với các ký tự tiếng Anh
     - ISO-8859-3: Các ký tự Tiếng Thổ Nhĩ Kỳ với các ký tự khác
   - Đã xác định là quá phức tạp/không đồng nhất
5. Unicode
   - Định nghĩa mỗi ký tự thành một số (điểm mã)
   - Ban đầu được mã hóa bằng 2 byte, cho phép có 16.000 ký tự có thể có (gọi là UCS-2)
   - Một lần nữa, số lượng ký tự trong tiêu chuẩn này đã vượt quá
   - UTF-16 được giới thiệu để cho phép mọi ký tự trên 16.000 được biểu diễn bằng cặp byte thứ hai
   - UTF-8 cho phép sử dụng từ 1 đến 6 byte để mã hóa một ký tự, với độ dài của ký tự được mã hóa vào các bit theo thứ tự cao của số ký tự và duy trì sự tương thích đầy đủ với bảng mã ASCII 128 ký tự ban đầu
   - UTF là kiểu mã hóa thống trị
6. iconv
   - Một tiện ích được sử dụng để chuyển đổi giữa các bộ mã ký tự
   - -c: Xóa các ký tự không xác định
   - -f [loại]: Từ loại được chỉ định
   - -t [loại]: Tới loại được chỉ định
   - -l: Liệt kê tất cả các loại mã hóa có sẵn
     - Ví dụ: iconv -c -f ASCII -t MACCYRILLIC VNCHOWTO > VNCHOWTO.new.cyrillic
       - Sẽ xóa bất kỳ ký tự không xác định nào trong luồng file từ VNCHOWTO và chuyển đổi từ mã hóa ASCII sang mã hóa MACCYRILLIC, ghi vào 1 file mới VNCHOWTO.new.cyrillic
   - **Lưu ý**: Đây không phải là công cụ dịch ngôn ngữ, chỉ là công cụ dịch mã hóa ký tự

Chủ đề 108 - Dịch vụ Hệ thống Cần thiết

108.1 Duy trì Thời gian Hệ thống (Đồng hồ Phần cứng, Đồng hồ Hệ thống và NTP) (Hardware Clock, System Clock and NTP)

1. Đồng hồ hệ thống(System clock)

    • Một module nhân kernel (kernel module) phản hồi với các giá trị thời gian khi ghi file hoặc được truy vấn về ngày/giờ.

2. date
    • Hiển thị ngày và giờ hiện tại cùng với múi giờ đã cấu hình.
    • Bạn có thể điều khiển định dạng đầu ra bằng ký tự + tiếp theo là các hướng dẫn định dạng đặc biệt.
    • %d: Ngày hai chữ số
    • %H: Giờ hai chữ số, định dạng 24 giờ
    • %m: Tháng hai chữ số
    • %M: Phút hai chữ số
    • %s: Số giây kể từ epoch (thời điểm đặc biệt trong lịch Unix)
    • %Y: Năm bốn chữ số
    • %z: Độ lệch múi giờ
        • Ví dụ: date +"%H:%M-%m%d%Y-%z"
            • Sẽ hiển thị thời gian như "23:11-04212017-0500"
    • -u: Hiển thị thời gian UTC của hệ thống
    • Cũng có thể được sử dụng để thay đổi ngày/giờ bằng cách bỏ đi ký tự + trước giá trị
        • Định dạng là MMDDhhmm[[CC]YY][.ss]]
        • MM: Tháng hai chữ số
        • DD: Ngày hai chữ số
        • hh: Giờ hai chữ số, định dạng 24 giờ
        • mm: Phút hai chữ số
        • CC: Thế kỷ hai chữ số (tùy chọn)
        • YY: Năm hai chữ số (tùy chọn)
        • .ss: Giây hai chữ số (tùy chọn)
3. Đồng hồ phần cứng (Hardware clock)
    • Đồng hồ nằm trên bo mạch (main board) của máy tính (hoặc trên bo mạch của máy ảo), thường được đặt và duy trì bởi pin khi máy tính tắt nguồn điện.
4. Độ lệch (Drift)
    • Sự khác biệt thời gian giữa đồng hồ phần cứng và đồng hồ hệ điều hành.

5. hwclock

- Cho phép làm việc trực tiếp với đồng hồ phần cứng.
- Hiển thị ngày/giờ của đồng hồ phần cứng.
- Lưu ý: Đồng hồ phần cứng không có ý thức về múi giờ.
  - **Ví dụ**: hwclock - trả về kết quả tương tự như "Mon 19 Jun 201709:13:54 PMUTC - 0.965537 giây".
    - Giá trị offset ở cuối là khoảng thời gian từ khi lệnh được thực thi và đồng hồ được đọc cho đến khi giá trị được hiển thị.
- Bạn có thể đồng bộ hóa đồng hồ phần cứng và Linux theo hai cách:
  - **Ví dụ:** hwclock -w (hoặc --systohc)
    - Sẽ đồng bộ hóa đồng hồ hệ thống với đồng hồ phần cứng.
  - **Ví dụ**: hwclock -s (hoặc --hctosys)
    - Sẽ đồng bộ hóa đồng hồ phần cứng với đồng hồ hệ thống.

6. /etc/adjtime
   - Chứa các giá trị theo dõi hiệu chỉnh của đồng hồ và giá trị cuối cùng để hiển thị thời gian theo múi giờ địa phương hoặc UTC.
   - Nếu file không tồn tại, không có hiệu chỉnh nào đã được thực hiện và thời gian sẽ mặc định là UTC.
7. ntp / ntpd
   - Có thể cần cài đặt gói ntp cho các tiện ích ở đây.
   - Giao thức thời gian mạng(Network Time Protocol) là dịch vụ daemon của giao thức thời gian mạng.
   - Cho phép bạn xác định một nhóm máy chủ mạng được đồng bộ hóa với một mạng lưới máy chủ thời gian phân tán trên internet.
     - Các máy chủ nhận cập nhật thời gian từ một đồng hồ "tham chiếu" (như viện quan sát Hải quân ở Hoa Kỳ) được gọi là "máy chủ stratum 1".
8. ntpdate
   - Cho phép bạn đặt đồng hồ theo máy chủ NTP được chỉ định.
9. pool.ntp.org
   - Trang web hiển thị tên các máy chủ NTP public.
   - Các bí danh(Aliases) thông thường cho cấu hình NTP: 0.pool.ntp.org đến 3.pool.ntp.org.
10. /etc/ntp.conf
    - Nhóm máy chủ NTP.
    - Xác định file drift để theo dõi hiệu suất đồng hồ.
11. /var/lib/ntp/drift
    - File theo dõi sự lệch thời gian của hệ thống Linux
    .
12. ntpq
    - Truy vấn một máy chủ NTP để xem thông tin thống kê và kết nối với hệ thống cục bộ theo mặc định.
    - Dấu nhắc lệnh đặc biệt
      - Peers: Các máy chủ thời gian đã được liên kết đến
      - Associations: Thông tin chi tiết hơn về mỗi máy chủ

108.2 System Logging

1. syslog (có thể là rsyslog, klogd)
   • Giao thức đơn giản để ghi nhật ký(log) thông điệp hệ thống
   • Các ứng dụng có thể truy cập vào syslog thông qua lệnh thư viện syslog hoặc qua tiện ích dòng lệnh logger
   • Daemon (syslogd) sẽ xử lý các thông điệp, các cấp độ ghi nhật ký có thể là:
      • 0: Khẩn cấp (Emergency)
      • 1: Cảnh báo (Alert)
      • 2: Nghiêm trọng (Critical)
      • 3: Lỗi (Error)
      • 4: Cảnh báo (Warning)
      • 5: Thông báo (Notice)
      • 6: Thông tin (Info)
      • 7: Gỡ lỗi (Debug)
   • Ghi nhật ký ở một cấp độ cụ thể sẽ giới hạn thông điệp được ghi nhật ký ở cấp độ đó hoặc cao hơn (cấp độ ghi nhật ký 3 có nghĩa là sự kiện 0, 1, 2, 3 được ghi nhật ký trong khi các sự kiện 4 và thấp hơn bị bỏ qua)
2. rsyslog
   • Phiên bản "nhanh" thay thế của syslog
3. syslog-ng
   • Phiên bản "thế hệ tiếp theo" thay thế của syslog
4. "facilities" ghi nhật ký
   • Ghi nhật ký cụ thể được liên kết với các mục khác nhau (mô tả những gì được ghi nhật ký)
      • kern: Các thông điệp kernel (Kernel messages)
      • user: Các thông điệp cấp người dùng (User level messages)
      • mail: Máy chủ email (Mail server)
      • daemon: Dịch vụ/daemon (Services/daemons)
      • auth: Nhật ký bảo mật (public) (Security logs)
      • syslong: Các thông điệp syslog nội bộ (Internal syslog messages)
      • lpr: In ấn (Printing)
      • cron: Lập lịch công việc (Job scheduling)
      • authpriv: Nhật ký bảo mật (riêng tư) (Security logs)
      • local0-7: Định nghĩa bởi người dùng (có sẵn 8 cái khác nhau)(User definable)

5. /var/log/message
   • Nơi chứa tất cả các thông điệp nhật ký (trừ mail)
6. /var/log/maillog
   • Các thông điệp emails được ghi vào đây

7. logger
   • Cho phép bạn sử dụng một lệnh ghi một thông điệp vào /var/log/messages
   • Nhấn CTL+D để kết thúc và ghi tin nhắn
   • -i: Chuyển thông tin bổ sung đến syslog
8. /etc/syslog.conf

- file cấu hình syslog
- Xác định nơi các facilities cụ thể sẽ ghi nhật ký

9. systemd
   - Sử dụng hệ thống ghi nhật ký riêng gọi là journal (với journald là daemon cho nó)
   - Được áp dụng trên hầu hết các bản phân phối Linux dựa trên systemd hiện đại
   - Chủ yếu, khác biệt nằm ở việc ghi nhật ký được thực hiện vào một file nhị phân chứ không phải file văn bản thuần túy, cho phép bạn truy vấn dữ liệu siêu dữ liệu, chi tiết dòng lệnh, PIDs, các files nhị phân và đặc quyền bảo mật (một số trong số đó không có sẵn trong file văn bản thuần)
   - Bởi vì nó là một phần của hệ thống quản lý dịch vụ trong Linux, tất cả các thông điệp của daemon được tự động ghi nhật ký thay vì syslog phiên bản sysvinit trong đó mỗi dịch vụ chịu trách nhiệm về cách và những thông điệp nào được ghi vào nhật ký
10. /var/log/journal
    - file nơi lưu trữ thông tin ghi nhật ký của systemd
11. journalctl
    - Lệnh được sử dụng để xem file nhật ký journal đã đề cập trên console
    - -f: Cho phép bạn theo dõi khi có các thông điệp nhật ký mới được ghi
    - SYSLOG_IDENTIFIER=[giá trị]: Cho phép bạn thêm bộ lọc để chỉ hiển thị các mục nhật ký phù hợp với tiêu chí
    - -u [dịch vụ]: Chỉ hiển thị các thông điệp được ghi nhật ký từ dịch vụ được chỉ định
    - -o verbose: Hiển thị nhiều thông tin hơn về dịch vụ hoặc bộ lọc được yêu cầu
    - -e: Di chuyển đến cuối nhật ký
    - -x: Thêm thông tin để giải thích ngữ cảnh trong đó một sự kiện hoặc lỗi được ghi nhật ký xảy ra (như trong trường hợp lỗi khởi động dịch vụ)
12. /etc/systemd/journald.conf
    - File cấu hình cho journ
    - Các thiết lập thông thường cho kích thước nhật ký và xem xét việc chuyển tiếp ghi nhật ký đến syslog cũng như tương đương đã được cài đặt.

13. Log rotation(xoay vòng nhật ký)

    - Định kỳ sao lưu, nén và/hoặc xóa các files nhật ký đáp ứng tiêu chí nhất định để quản lý không gian đĩa và hiệu suất I/O.

    - /etc/logrotate.conf (và bất kỳ thứ gì trong /etc/logrotate.d)

    - Cấu hình chính cho việc xoay vòng nhật ký (cài đặt mặc định và file hệ thống để xoay vòng)

    - Mỗi file trong logrotate.d thêm hoặc ghi đè các thiết lập lên cài đặt mặc định trong file cấu hình

      - **Lưu ý**: Các files trong logrotate.d thường được thêm/duy trì bởi trình quản lý gói(package manager) là một phần của quá trình cài đặt(install)/cập nhật(update)/xóa(removal) gói

• Các files đang mở trong quá trình xoay vòng nhật ký được xử lý theo một trong ba cách:

• Di chuyển nhật ký, tạo tín hiệu đóng file và mở lại nhật ký; không phải ứng dụng nào cũng hỗ trợ phương pháp này

• Khởi động lại ứng dụng/dịch vụ sau khi di chuyển file nhật ký, nhật ký sẽ bắt đầu ghi vào một file mới "trống"; có thể gây vấn đề nếu khởi động lại dịch vụ ảnh hưởng đến người dùng hoặc kết nối

• Sao chép nhật ký và sau đó cắt ngắn nhật ký hiện tại ngay tại chỗ; tốn kém về hiệu suất ổ đĩa và các mục nhập nhật ký có thể bị mất trong quá trình giao dịch

108.3 Cơ bản về Mail Transfer Agent (MTA)

(Tổng quan về Email)

1. Máy chủ mail
   • Sendmail
     • Máy chủ mail kiểu cũ, có thể là "tiêu chuẩn" mà các bản phân phối Linux sẽ cài đặt. Có điểm yếu dễ bị tấn công bảo mật và được biết đến là khó cấu hình.
   • Postfix
     • Ban đầu được phát triển bởi IBM, đã thay thế Sendmail như máy chủ thư mặc định trên các bản phân phối Linux hiện đại. Tích hợp hỗ trợ các luật để giảm thư rác và phân tách các tiến trình vì mục đích bảo mật.
   • Qmail

     • Là một sự thay thế an toàn cho Sendmail. Các tiến trình riêng biệt cho phép cải thiện bảo mật và cô lập và cấu hình đơn giản hơn, các cải thiện này giúp Qmail trở thành một sự lựa chọn tốt. Tuy nhiên, gần đây việc phát triển nó đã kém hơn.

   • Exim

     • Giống như Sendmail với tính chất và cấu hình "monolithic" hơn, mặc dù nó được biết đến với lịch sử bảo mật tốt hơn. Cấu hình đơn giản hơn và đôi khi được coi là máy chủ mail mặc định trong các phiên bản cũ của bản phân phối Debian.

2. SMTP
   • Simple Mail Transfer Protocol: Đây là một giao thức định nghĩa cách emails được chuyển tiếp và lưu trữ và là một phần của tầng ứng dụng TCP/IP cũng như các quy tắc cấu hình mà các ứng dụng emails tuân thủ theo.
3. MUA
   • Mail User Agent: Đây là các ứng dụng nào bạn sử dụng để soạn và gửi email (Thunderbolt, Evolution, SquirrelMail, v.v.).

4. MSA
   • Mail Submission Agent: Hoạt động như một trung gian hoặc cổng thông tin giữa MUA và MTA để bắt đầu quá trình chuyển tiếp email.
5. MTA
   • Mail Transfer Agent: Nhận emails từ MUA và gửi nó (nếu cần) đến máy chủ mail đích (MTA khác nếu đây không phải là đích). Có nhiều máy chủ MTA trong Linux (Postfix, mà chúng ta sẽ sử dụng, Sendmail và nhiều loại máy chủ mail khác).
6. MDA
   • Mail Delivery Agent: Nhận email từ MTA và sau đó chuyển nó đến mail spool cục bộ để người dùng có thể lấy emails về bằng các loại ứng dụng thư điện tử. Đôi khi MTA cũng có thể hoạt động như MDA, nhưng (ví dụ như procmail) chúng là các ứng dụng độc lập cũng có thể dùng thể lọc emails (như chống spam).
7. POP
   • Post Office Protocol: Được MUAs sử dụng bởi để lấy email (sẽ được đề cập chi tiết hơn sau).
8. IMAP
   • Internet Message Access Protocol: Được MUAs sử dụng để lấy email (sẽ được đề cập chi tiết hơn sau).
9. MX Record
   • Đây là các bản ghi DNS mail. Các bản ghi này được sử dụng bởi MTAs để xác định máy chủ mail có thẩm quyền cho bất kỳ email nào.

108.3 Mail Transfer Agent (MTA) Basics (Tạo bí danh)

1. Mặc dù cấu hình và cài đặt máy chủ mail KHÔNG có trong câu hỏi của kỳ thi, tuy nhiên bạn nên làm theo các bước cài đặt máy chủ mail trên máy chủ của bạn; chỉ cần cài đặt sendmail và khởi động dịch vụ.
2. /etc/aliases
   • Cho phép gọi một người dùng bằng tên/khác
   • **Ví dụ (ghi trong file):** sysadmin: root
   • Sẽ chỉ ra là nếu một email được gửi đến máy chủ mail cho tài khoản sysadmin sẽ được chuyển tới root
   • Định dạng của một bí danh (alias): alias: account[,another,another]
   • Dấu ngoặc vuông là tùy chọn, xác định danh sách các tài khoản mà một bí danh đề cập đến là một danh sách được phân cách bằng dấu phẩy
   • **Lưu ý**: Một bí danh có thể là bất kỳ tên nào, có thể được sử dụng để gửi email của một tài khoản tới một tài khoản khác
   • Ví dụ: root: user,user1
   • Gửi email của root tới user và user1
   • Cũng có thể chấp nhận địa chỉ email là một tài khoản để gửi mail nội bộ (nếu máy chủ mail được cấu hình)
   • **Ví dụ:** sysadmin: user@abc.com
   • Sẽ gửi email của sysadmin tới địa chỉ user@abc.com

- Email cũng có thể được gửi tới một file từ một bí danh
  - **Ví dụ**: sysadmin: /root/sysadmin.email
    - Sẽ gửi tất cả email tới file sysadmin.email trong thư mục của root user
- Có thể gửi tới một kịch bản/ứng dụng
  - **Ví dụ**: support: | /opt/HelpDesk/create_ticket.sh
    - Sẽ gửi email và gây lên việc thực thi một kịch bản (lưu ý kịch bản phải được tiền đề bằng ký tự ống |)

3. /etc/aliases.db
   - file cơ sở dữ liệu mà MDA cục bộ của bạn sẽ đọc để xác định nơi gửi email; nó phải được cập nhật khi có bất kỳ thay đổi nào được thực hiện với bí danh trong Linux
4. newaliases
   - Lệnh cập nhật aliases.db với các thay đổi trong file aliases ở trên
   - Sẽ hiển thị lỗi nếu có, thành công sẽ không trả về kết quả
   - Nếu bạn thay đổi aliases mà không chạy lệnh này, thông báo sẽ được ghi lại trong /var/log/messages là cơ sở dữ liệu cũ hơn file aliases
5. forward
   - file trong thư mục home cho người dùng để định nghĩa các quy tắc chuyển tiếp các emails riêng của họ
   - Không cần chỉ định bí danh, vì nó thuộc về tài khoản mà nó tồn tại trong; chỉ cần chỉ định tài khoản để chuyển tiếp emails tới
6. mailq
   - Lệnh hiển thị hàng đợi email
7. var/spool/mqueue
   - Các emails đang chờ được gửi đi
   - Chỉ tồn tại nếu có emails trong hàng đợi; trên một máy chủ được cấu hình đúng, nó nên là trống
8. /var/log/maillog
   - Ghi lại tất cả các thông điệp thư điện tử khi chúng đi vào và ra khỏi Linux

108.4 Quản lý máy in và in ấn (CUPS và lpr)

1. lpd
   • Lệnh dịch vụ in truyền thống trước khi có CUPS, là phương pháp mặc định để gửi đầu ra đến máy in/thiết bị in ấn
   • **Lưu ý**: Vì mục đích thực hành, bạn sẽ cài đặt lpd, cài đặt gói cups-lpd để chúng hoạt động với các lệnh cũ và cài đặt cups-pdf để cài đặt máy in để làm việc
2. ipp
   • Giao thức in internet (dựa trên web)
3. CUPS
   • Hệ thống in chung của HĐH Unix
   • Kết hợp của các công cụ chuyển đổi, bộ lọc và "trình điều khiển(driver) máy in" mà tài liệu có thể được gửi đến khi đầu ra đến các thiết bị in ấn, các kịch bản để tạo tài liệu định dạng SVG, PDF hoặc định dạng khác
4. cupsd

- Tiến trình nền (daemon) cho CUPS
- CUPS hoạt động trên cổng 631 và có thể truy cập thông qua trình duyệt local của bạn (http://localhost:631)

5. /etc/cups
   - Thư mục cấu hình cho CUPS
   - Danh sách các files cấu hình:
     - classes.conf: Cấu hình định nghĩa các lớp
     - cupsd.conf: file cấu hình chính cho tiến trình nền (daemon)
     - cupsd.conf.default: file cấu hình mặc định mẫu để khôi phục dự phòng
     - printers.conf: Cấu hình của từng máy in trên hệ thống
     - ppd: Thư mục PPD (file trình điều khiển máy in) trên mỗi máy in trên hệ thống
6. cupsreject
   - Được sử dụng để đặt máy in từ chối tất cả các công việc in đã được gửi đến nó. Bằng cách chỉ định tên máy in, lệnh này sẽ chặn tất cả các công việc in tiếp theo được gửi đến máy in đó.
7. cupsaccept
   - Bằng cách chỉ định tên máy in, lệnh này sẽ đặt máy in đó chấp nhận tất cả các công việc in đã được gửi đến nó.
8. cupdisable
   - Vô hiệu hóa máy in đã chỉ định (nhưng vẫn chấp nhận công việc in, chỉ giữ trong hàng đợi).
9. cupsenable
   - Kích hoạt máy in đã chỉ định.
10. cupsctl
    - Sử dụng để điều khiển cấu hình CUPS, chạy mà không có tùy chọn, hiển thị cấu hình hiện tại.
11. lpadmin
    - Tiện ích tương thích với lp để quản lý hàng đợi lp (nhớ là plugin tương thích với CUPS, do đó sẽ áp dụng cho toàn bộ cấu hình in cục bộ).
    - Có thể cấu hình để thêm một máy in test bằng lệnh sau:
      - lpadmin -p MyTestPrinter -E -v /dev/null -m raw
12. lpstat
    - Hiển thị cấu hình lp hiện tại (máy in và mặc định).
    - -p: Hiển thị máy in
    - -d: Hiển thị mặc định
    - -a: Hiển thị tất cả hàng đợi in và trạng thái
    - -r: Trạng thái dịch vụ
    - -s: Tổng quan cấu hình hệ thống
    - -t: Tổng quan chi tiết cấu hình hệ thống
13. lpoptions
    - Cho phép đặt các tùy chọn máy in từ dòng lệnh.
    - -d [máy in]: Đặt máy in mặc định thành giá trị đã chỉ định.
    - -p [máy in]: Xử lý máy in đã chỉ định.
    - -l: Liệt kê các tùy chọn cho máy in/hàng đợi đã chỉ định.
      - **Ví dụ**: lpoptions -p CUPS-PDF -l
        - Sẽ hiển thị tất cả các tùy chọn cho máy in CUPS-PDF.

14. lp
    - Các công cụ dòng lệnh (cổ điển) để in.
        - **Ví dụ:** echo "my test print job" | lp
15. lpr
    - Là các công cụ dòng lệnh (cổ điển) để in tài liệu.
        - **Ví dụ:** echo "công việc in kiểm tra của tôi" | lpr
        - Sẽ in kết quả của lệnh echo vào máy in mặc định.
        - -P [printer]: Máy in đích (nếu không phải máy in mặc định).
        - -#[#]: In số bản sao được chỉ định.
16. lpq
    - Hiển thị hàng đợi in và công việc in.
    - -a: Hiển thị tất cả công việc in cho tất cả máy in.
    - -P [printer]: Hiển thị công việc in cho máy in được chỉ định.
17. lprm
    - Cho phép bạn xóa công việc in.
    - **Lưu ý**: Chạy lệnh mà không có một job ID sẽ xóa công việc in đầu tiên trong hàng đợi.
    - -P [printer]: Làm việc với máy in được chỉ định.
    - -a: Xóa tất cả các công việc in trong hàng đợi.

Chủ đề 109 - Cơ bản về Mạng

109.1 Cơ bản về Giao thức Internet

1. IP (Internet Protocol)
    - Một phương pháp xác định duy nhất địa chỉ (đích) cho một hệ thống cụ thể. Có hai phiên bản chính:
        - IPv4: Cấu trúc địa chỉ tiêu chuẩn gồm bốn "octet" chứa các số từ 0-255 cho mỗi octet (**ví dụ**: 192.168.72.211).
        - IPv6: Được thiết kế để thay thế IPv4, gồm một số thập lục phân 128 bit cho địa chỉ (**ví dụ**: 2DAF:FF40:0928:CD01:4433:00DD:0988:FFFF).
2. TCP (Transmission Control Protocol)
    - Phương pháp mà tất cả các giao dịch giữa các địa chỉ IP (bất kỳ phiên bản nào) được truyền thông. Định nghĩa một hệ thống truyền và xác nhận để đảm bảo lưu lượng dữ liệu đến và có thể được lắp ráp theo đúng thứ tự.
3. UDP (User Datagram Protocol)

- Thường được coi là "bổ sung" cho IP, nhưng là một kết nối "không lưu trạng thái". Không có kiểm tra lỗi hoặc gửi lại các gói tin, ngay cả khi gửi gói tin không thành công.

4. ICMP (Internet Control Message Protocol)

- Được thiết kế cho các thiết bị mạng (bộ định tuyến(routers), switch thông minh, tường lửa(firewall), vv.) để gửi các thông báo lỗi. Ngoài ra, nó cũng có thể thực hiện các truy vấn sẵn có của dịch vụ mạng (như trong trường hợp sử dụng lệnh ping để kiểm tra xem một địa chỉ có đáp ứng yêu cầu hay không).

5. Phạm vi Lớp IP
   - RFC 1918 chỉ định năm phạm vi. Các giá trị trong mỗi phạm vi xác định tổng số lượng các hosts trong mỗi lớp (địa chỉ):
     - Lớp A: Từ 1 đến 126 - có 16.777.214 địa chỉ hosts
     - Lớp B: Từ 128 đến 191 - có 65.534 địa chỉ hosts
     - Lớp C: Từ 192 đến 223 - có 254 địa chỉ hosts
     - Lớp D: Từ 224 đến 239 - dùng cho multicast và không sử dụng cho địa chỉ hosts
     - Lớp E: Từ 240 đến 254 - dành cho việc sử dụng trong tương lai
6. Mặt nạ mạng(Network Mask)
   - Định nghĩa một mạng logic (gọi là mạng con subnet) chỉ ra điểm bắt đầu và kết thúc của một dải địa chỉ.
7. Phạm vi
   - Mỗi phạm vi lớp địa chỉ có một mặt nạ mạng/mạng con tương ứng.
     Số bit (được chỉ định bởi /) được gọi là ký hiệu Ghi chú địa chỉ mạng (CIDR).
     - Lớp A: 255.0.0.0 (hoặc /8)
     - Lớp B: 255.255.0.0 (hoặc /16)
     - Lớp C: 255.255.255.0 (hoặc /24)
8. Cổng mạng (Gateway)
   - Được sử dụng để kết nối mạng cục bộ với mạng bên ngoài, chẳng hạn như kết nối mạng nội bộ với Internet. Gateway có vai trò chuyển tiếp gói tin giữa các mạng khác nhau và thực hiện chức năng định tuyến để định địa chỉ đích của gói tin.
9. Địa chỉ Broadcast (phát sóng)
   - Địa chỉ Broadcast
   - Mỗi mạng đều có một địa chỉ Broadcast, hoạt động trên số địa chỉ là 255.
   - **Ví dụ:** Mạng 192.168.1.0/24
     - 192.168.1.255 là địa chỉ Broadcast
   - Đây là cách mà một mạng IP gửi lưu lượng mà có thể được nhìn thấy bởi tất cả các máy trên mạng.

10. Cổng/Dịch vụ phổ biến (trong kỳ thi)

- Có một số trong các cổng/dịch vụ sau có thể xuất hiện trong kỳ thi:

• 20/21: FTP

• 22: SSH

• 23: Telnet

• 25: SMTP

• 53: DNS

• 80: HTTP

• 110: POP3

• 123: NTP

• 139: NETBIOS

• 143: IMAP

• 161/162: SNMP

• 389: LDAP

• 443: HTTPS

• 465: SMTPS

• 514: SYSLOG

• 636: LDAPS

• 993: IMAPS

• 995: POP3S

109.2 Cấu hình Mạng Cơ bản (Làm việc với Card mạng và định tuyến Routes)

1. nmcli
   • công cụ giao diện dòng lệnh cho NetworkManager
   • dev: Hiển thị thông tin thiết bị (liên quan đến phần cứng mạng)
   • con: Hiển thị thông tin cấu hình kết nối

2. ifconfig
   - Công cụ để xem tất cả các card mạng hoạt động (bao gồm cả card loopback)
   - -a: Hiển thị TẤT CẢ các card mạng (hoạt động hoặc không hoạt động)
   - Các trường quan trọng:
     - ether: Địa chỉ phần cứng (MAC), địa chỉ vật lý của card mạng 48 bit
     - inet: Địa chỉ mạng được gán cho card mạng đó
     - broadcast: Địa chỉ broadcast (phát sóng) cho mạng của hệ thống
     - netmask: Mặt nạ mạng hoặc thông tin đoạn mạng logic
   - **Lưu ý**: Đây là một lệnh mạng cổ điển đang bị thay thế bằng lệnh thế hệ mới
3. ifup
   - Kích hoạt giao diện mạng (card mạng) được chỉ định
4. ifdown
   - Tắt giao diện mạng(card mạng) được chỉ định
5. ip
   - Lệnh quản lý mạng và định tuyến thống nhất được thiết kế để thay thế chức năng của hầu hết các lệnh khác mà bạn cần biết cho kỳ thi
   - **Ví dụ**: ip addr show
     - Sẽ cung cấp cho bạn thông tin tương tự như lệnh ifconfig ở trên
6. file cấu hình card mạng
   - Red Hat/CentOS (v6)
     - /etc/sysconfig/network-scripts
       - Một thư mục chứa các tập lệnh chịu trách nhiệm cấu hình tất cả các interfaces trên hệ thống
         - Ví dụ: ifcfg-eth1
           - Chịu trách nhiệm cho cấu hình (IP tĩnh hoặc DHCP) thông tin địa chỉ cho interface ETH1 trên hệ thống Linux của bạn
       - Thay đổi cấu hình giao diện mạng được áp dụng bằng cách khởi động lại dịch vụ mạng
         - **Ví dụ**: service network restart
     - Debian/Ubuntu (không được kiểm tra trong kỳ thi LPIC)
       - /etc/network/interfaces
       - Được cấu hình bằng system-config-networking hoặc netcardconfig
       - Thay đổi cấu hình giao diện mạng được áp dụng bằng cách khởi động lại dịch vụ networking
       - Ví dụ: /etc/init.d/networking restart
7. Default gateway(Cổng mặc định)
   - Đích của TẤT CẢ lưu lượng mạng có đích không nằm trong mạng hệ thống mạng cục bộ HOẶC không có một static route định tuyến tĩnh phù hợp khác được cấu hình
   - Có thể được cấu hình trong /etc/sysconfig/network hoặc /etc/sysconfig/network-scripts/ifcfg-eth0 (trong đó # là số giao diện)
     - **Ví dụ**: GATEWAY=192.168.1.1
       - Sẽ cấu hình Default GW của Linux với địa chỉ IP 192.168.1.1
8. route
   - Hiển thị bảng định tuyến(routing table) hiện tại
   - Thêm/xóa định tuyến theo yêu cầu

- **Ví dụ**: route add default gw 192.168.1.1
  - Sẽ thêm một Default GW vào hệ thống đi đến 192.168.1.1
- **Ví dụ**: route add 192.168.10.211 lo
  - Sẽ định tuyến tất cả lưu lượng(traffic) trả về cho loopback adapter (giải pháp nhanh chóng chống lại một cuộc tấn công mạng DoS attack từ một địa chỉ IP duy nhất, trường hợp này IP là: 192.168.10.211)
- add: Thêm một gateway/route/destination
- default: Từ khóa để thêm/xóa một Default GW
- gw: Viết tắt của gateway, tất cả lưu lượng không khớp với các quy tắc khác được định tuyến đến đây
- [Địa chỉ IP]: Địa chỉ IP của định tuyến/ gateway/ mạng

9. IP forwarding
   - Khả năng cho phép máy chủ của bạn chuyển tiếp gói tin đến vị trí khác và phản hồi
   - Cho phép Linux của bạn hoạt động như một bộ định tuyến router
   - Có hai phương pháp để kích hoạt:
     1. echo 1 > /proc/sys/net/ipv4/ip_forward
     2. sửa /etc/sysctl.conf và thêm net.ipv4.ip_forward=1
        - **Lưu ý**: Phương pháp một không phải là vĩnh viễn nhưng sẽ có hiệu lực ngay lập tức, phương pháp hai yêu cầu khởi động lại (hoặc kết hợp với phương pháp một)
10. Gia hạn địa chỉ DHCP lease
    - dhcpd -k: Dừng và khởi động lại tiến trình để nhận địa chỉ IP mới
    - dhclient: Sau khi khởi động lại dịch vụ mạng(network services), chạy để nhận địa chỉ IP mới
    - pump: Công cụ cũ, dùng để nhận một địa chỉ DHCP mới
11. hostnamectl
    - Lệnh được sử dụng để cấu hình tên máy cục bộ một cách vĩnh viễn

109.3 Sửa xử lý lỗi cơ bản trong mạng

1. ping
   - Công cụ để kiểm tra phản hồi từ một địa chỉ mạng cụ thể
   - -n [#]: Ping một số lần chỉ định
1. netstat
   - Có thể hiển thị kết nối mạng, bảng định tuyến, thống kê card mạng, vv.
   - -a: Hiển thị tất cả các socket trên giao diện mạng đang hoạt động
   - -c: Làm mới thống kê mỗi 1 giây
   - -p: Hiển thị tên và PID cho mỗi socket
   - -t: Hiển thị thống kê TCP
   - -r: Hiển thị bảng định tuyến
   - -n: Không phân giải tên (chỉ hiển thị IP)
3. traceroute (chỉ có root có quyền chạy lệnh này)
   - Công cụ để xác định khoảng cách (theo số bước nhảy) giữa hệ thống của bạn và điểm cuối mong muốn, cũng như thời gian phản hồi của mỗi bước trên đường đi

- -n: Không phân giải tên cho mỗi bước nhảy, chỉ hiển thị IP

4. traceroute6
   - Tương đương IPv6 của traceroute
5. tracepath (tất cả người dùng)
   - Công cụ để xác định khoảng cách (theo số bước nhảy) giữa hệ thống của bạn và điểm cuối mong muốn, cũng như thời gian phản hồi của mỗi bước trên đường đi
   - -n: Không phân giải tên cho mỗi bước nhảy, chỉ hiển thị IP
6. tracepath6
   - Tương đương IPv6 của tracepath
7. ping6
   - Lệnh ping cho IPv6

109.4 Cấu hình DNS phía máy khách

1. /etc/hosts
   - File cục bộ chứa tên liên kết với một địa chỉ IP
2. /etc/resolv.conf
   - Xác định máy chủ DNS và tên miền của hệ thống để phân giải tên miền
3. /etc/nsswitch.conf
   - Xác định thứ tự các truy vấn phân giải tên cục bộ so với truy vấn phân giải tên miền từ xa
     - Ví dụ: hosts: files dns
       - Dòng cấu hình trên sẽ cho phép Linux kiểm tra /etc/hosts trước khi kiểm tra các máy chủ DNS có trong /etc/resolv.conf
4. **Lưu ý**: Các lệnh sau đây yêu cầu gói bind-utils
5. Host
   - Lấy thông tin DNS về máy chủ chỉ định (thực hiện các truy vấn DNS)
6. getent
   - hosts [name]: Lấy thông tin về tên máy chủ chỉ định (sẽ đọc /etc/nsswitch.conf để xác định thứ tự tìm kiếm)
7. dig
   - [server] [domain] [loại bản ghi]: Tất cả tùy chọn trừ domain
   - server: Chỉ định một máy chủ DNS để sử dụng
   - domain: Miền cụ thể để truy vấn
   - record type: Các loại bản ghi khác nhau (ví dụ: NAME, CNAME, MX, vv.)

110.1 Thực hiện các nhiệm vụ Quản trị Bảo mật

1. root
   - Tài khoản quản trị hệ thống
   - Thực hành bảo mật tốt nhất là không đăng nhập trực tiếp vào hệ thống với quyền root, mà là "trở thành" root
   - Hai phương pháp:

1. su: Nhập mật khẩu root và sau đó trở thành root (su = superuser)
2. sudo su: Nhập mật khẩu của bạn và, miễn là bạn có đặc quyền sudo (sudo = superuser do), sau đó bạn sẽ trở thành root

2. su
- Super user
- **Ví dụ**: su root
- Trở thành người dùng root, nhưng các kịch bản shell đăng nhập của user root(.bash_login, .bashrc, vv.) sẽ không được thực thi, vì vậy bạn sẽ không thừa hưởng các giá trị của môi trường shell của user root
- Ví dụ: su - root
- Trở thành root và tất cả các kịch bản shell đăng nhập sẽ được thực thi và môi trường và cài đặt đầy đủ sẽ được sử dụng
- Cũng có thể được thực hiện với su -l root
- Áp dụng cho cả người dùng thường (miễn là bạn biết mật khẩu tương ứng)
- **Lưu ý:** Root có thể su sang bất kỳ người dùng nào mà không cần phải biết mật khẩu của người dùng đó

3. visudo
- Chế độ chỉnh sửa đặc biệt cho VI cho phép chỉnh sửa và kiểm tra cú pháp/lỗi của file /etc/sudoers
- **Lưu ý**: Trình soạn thảo này được sử dụng có thể được thay đổi bằng cách đặt biến môi trường EDITOR thành bất kỳ trình chỉnh sửa văn bản khác có sẵn

4. /etc/sudoers
- File liệt kê người dùng có thể thực thi các lệnh với đặc quyền root
- Định dạng yêu cầu người dùng được liệt kê (hoặc một nhóm nếu toàn bộ nhóm đó phải có đặc quyền root) cùng với các quyền của họ trong Linux
- **Ví dụ**: user ALL=(ALL) ALL
- Sẽ cung cấp quyền sudo cho tài khoản người dùng cho tất cả các lệnh được sử dụng với đặc quyền root
- **Ví dụ:** user ALL=(ALL) /bin/systemctl
- Sẽ giới hạn người dùng chỉ có thể chạy systemctl với đặc quyền sudo (để có thể khởi động lại các dịch vụ services)

5. find
- Ngoài các lệnh thường để tìm kiếm các files, thư mục, v.v., có thể sử dụng để tìm kiếm các files có các quyền cụ thể hoặc do người dùng cụ thể sở hữu

- Được sử dụng để khôi phục từ việc thay đổi quyền do sơ suất (hoặc độc hại)

- -user [user]: Tìm các files thuộc sở hữu của người dùng được chỉ định

- -perm [permissions]: Tìm các files có các quyền được chỉ định

- Có thể bao gồm các quyền "bit đặc biệt"

- +: Tìm bất cứ thứ gì phù hợp với số đầu tiên

- -: Tìm kiếm chính xác khớp cho số đầu tiên

- -type [type]: Tìm kiếm các files, liên kết, thiết bị, thư mục
- -exec [commands] {} ;: Thực hiện các lệnh đã chỉ định trên các files được tìm thấy
- **Ví dụ**: find / -user root -perm 0777 -type f -exec ls -al {} ; | mail -s "Files owned by root with world rwx permissions" root
- Lệnh trên sẽ tìm kiếm tất cả các files thuộc sở hữu của người dùng root với tất cả các quyền đọc/ghi/thực thi mà cho phép tất cả các người dùng có thể thực thi , sau đó gửi email danh sách đó với danh sách hiển thị của tất cả các files cho người dùng root với chủ đề(subject) được chỉ định

6. Các mục khác có thể được sử dụng để điều tra/bảo mật hệ thống (được bao gồm trong Kỳ thi LPIC-1)
   - passwd, fuser, lsof, chage, usermod, ulimit, w, /etc/inittab

110.2 Thiết lập Bảo mật Máy chủ

1. Quản lý dịch vụ
   - Xác định cách/khi nào một dịch vụ(service) đang chạy (luôn chạy, chạy theo yêu cầu hoặc bị vô hiệu hóa(disabled))
2. **Lưu ý:** Cả hai đều không được cài đặt mặc định trên các bản phân phối Linux hiện đại
3. inetd
   - Tiến trình/dịch vụ cũ dùng để cung cấp các dịch vụ hệ thống "theo yêu cầu" hoặc khi cần thiết
   - /etc/inetd.conf: File cấu hình chính chứa một dòng duy nhất cho mỗi kiểm soát dịch vụ
     - **Ví dụ**: telnet stream tcp nowait root /usr/sbin/in.telnetd in.telnetd
       - Cấu hình mẫu cho telnet, theo thứ tự các trường sau:
         - Tên dịch vụ: Tên của dịch vụ cần kiểm soát (telnet)
         - Loại socket: Các giá trị có thể là stream, dgram, rdm, seqpacket (stream)
         - Giao thức: File /etc/protocols xác định các giá trị, thông thường là TCP hoặc UDP (tcp)
         - Wait/nowait: Chờ cho các dịch vụ đơn luồng, nowait cho các dịch vụ đa luồng (nowait)
         - Người dùng(user)/nhóm(group): Nhóm hoặc tài khoản người dùng mà dịch vụ sẽ chạy dưới dạng (root)
       - Đường dẫn đến lệnh: Đường dẫn đầy đủ đến ứng dụng/dịch vụ để chạy (/usr/sbin/in.telnetd)
       - Đối số: Bất cứ điều gì có thể được yêu cầu để hoàn thành cấu hình dịch vụ (in.telnetd)
4. xinetd
   - Thay thế cho inetd để cho phép kiểm soát dịch vụ một cách chi tiết hơn

- /etc/xinetd.conf: File cấu hình chính, bao gồm các files trong /etc/xinetd.d với mỗi dịch vụ được kiểm soát
  - Yêu cầu sẽ đi tới daemon, nó sẽ kiểm tra loại dịch vụ và cổng, sau đó quét file cấu hình dịch vụ thích hợp trong /etc/xinit.d
  - Các trường quan trọng:
    - cps = [#] [#]: Số đầu tiên giới hạn số kết nối mỗi giây đến dịch vụ, số thứ hai là độ trễ trước khi trả lời cho phép thêm kết nối
    - instances = [#]: Tổng số tiến trình daemon được phép chạy cùng lúc

5. Các dịch vụ thường
   - finger, imap, rsh/rsync, telnet
   - File mẫu rsync từ /etc/xinet.d

```
service telnet
{
  flags = REUSE
  socket_type = stream
  wait = no
  user = root
  server = /usr/sbin/in.telnetd
  log_on_failure += USERID
  disable = no
}
```

   - **Ghi chú**: Chủ đề thi phổ biến là cách chạy dịch vụ từ xinetd dưới tài khoản người dùng khác root (**gợi ý**: xem trường user trong mỗi file cấu hình dịch vụ)
     - service: Tên của dịch vụ
     - disable: yes (vô hiệu hóa(disable) và không chấp nhận kết nối) và no (không vô hiệu hóa và cho phép kết nối)
     - flags: Biến đổi theo dịch vụ
     - socket type: Các giá trị có thể là stream, dgram, rdm, seqpacket
     - wait/nowait: wait cho các dịch vụ đơn luồng, nowait cho các dịch vụ đa luồng
     - user/group: Nhóm hoặc tài khoản người dùng mà dịch vụ sẽ chạy dưới dạng
     - server_args: Đối số gửi tới dịch vụ
     - log_on_failure: Ghi lại các lỗi bao gồm USERID
     - Để kích hoạt, đặt disable = no và khởi động lại xinetd
6. TCP wrappers
   - Đối với inetd, chỉ sử dụng /etc/hosts.allow và /etc/hosts.deny như là các tham số của tcpd
   - Đối với xinetd, thư viện libwrap.a cho phép các dịch vụ sử dụng /etc/hosts.allow và /etc/hosts.deny
   - **Ví dụ** (định dạng của mỗi dòng): [service/daemon]: [host(s)]: [option]: [option]

```
in.telnetd: 10.1.10.0/24
```

     - Cho phép (hoặc từ chối) người dùng trong mạng 10.1.10.0/24 truy cập vào dịch vụ telnet (phụ thuộc vào file nơi nó tồn tại)

- Các phương pháp sử dụng để định nghĩa các chính sách:

- Từ chối mặc định: Từ chối tất cả các máy truy cập đến tất cả các dịch vụ (ALL:ALL)
  - Sử dụng như một phần dự phòng trong hosts.deny vì các kết quả khớp trong hosts.allow sẽ ghi đè lên
  - Cho phép mặc định: Tự động tin tưởng tất cả mọi người và cung cấp quyền truy cập đến tất cả mọi thứ
    - Mối nguy hiểm về mặt bảo mật rõ ràng
  - Kết hợp: Cho phép và từ chối theo cách lựa chọn

7. Thứ tự ưu tiên cho hosts.allow và hosts.deny
   - hosts.allow: Được đọc trước, nếu khớp với các dòng cấu hình trong file thì được cho phép và hosts.deny được bỏ qua (hoàn toàn)
     - Thay đổi trong hosts.allow/deny có hiệu lực ngay sau khi thay đổi file
   - Đọc tuần tự: Có nhiều mục nhập cho cùng một dịch vụ sẽ chỉ áp dụng mục nhập đầu tiên
   - Nếu các files không tồn tại, không có quy tắc nào được áp dụng
   - Có thể thêm các tùy chọn khác thay vì chỉ từ chối dịch vụ
     - **Ví dụ:** in.telnetd: 10.1.10.0/24 : twist /bin/echo "Service 404 - Service Not Found"

110.3 Bảo mật dữ liệu bằng mã hóa (Sử dụng GnuPG để mã hóa và giải mã Files)

1. gnupg
   - GnuPG Privacy Guard
   - Tạo khóa công khai(public key) và khóa riêng(private key) để mã hóa dữ liệu
2. gpg
   - Công cụ làm việc với các keys
   - --gen-key
     - Sẽ yêu cầu bạn chọn loại khóa (RSA thường là mặc định)
     - Theo các yêu cầu để cung cấp dữ liệu trong quá trình tạo cặp keys của bạn
   - Entropy: "hạt giống" được sử dụng để tạo ra keys mã hóa, nó liên quan đến tính ngẫu nhiên của một số sự kiện trên HĐH Linux của bạn (việc tạo nó đúng cách có thể mất thời gian lớn đối với các khóa lớn)
     - Tạo entropy (Thử để testing)
       - yum install rngd-tools
       - rngd -r /dev/urandom
   - -a: Tạo public key dạng văn bản để cung cấp
   - -o [filename]: file đầu ra
     - **Lưu ý**: Khi tạo keys, hãy chắc chắn rằng bạn đã đăng nhập vào hệ thống trực tiếp với tên người dùng và mật khẩu hoặc môi trường hoặc GPG sẽ không được cài đặt mà không cần cấu hình thủ công

- **Lưu ý:** Khi hoàn thành, tên key sẽ được liệt kê trên đầu ra hiển thị lên màn hình, **CHÚ Ý** giá trị đó

- --import [key]: Nhập key được chỉ định (public key từ người sẽ gửi file cho bạn)

- --list-keys: Liệt kê tất cả các keys đã nhập của bạn

3. Mã hóa một file để gửi cho người dùng khác
   - Nhập file public key từ người nhận dự định sẽ gửi file mã hóa cho người này
   - Giải mã file bằng cách sử dụng khóa đã nhập và passphrase cần thiết
     - **Ví dụ**: gpg file.gpg
       - Sẽ giải mã file với điều kiện đã nhập đúng key và bạn có mật khẩu passphrase được tạo ra cùng với nó

110.3 Bảo mật Dữ liệu với Mã hóa (SSH Keys - Public và Private - cho Kết nối Bảo mật)

1. ssh
   - Secure Shell
     - Các lệnh liên quan (cũng bảo mật): scp, ssh-agent, ssh-add
   - -l [user] [host]: Đăng nhập dưới tên người dùng được chỉ định vào máy chủ
   - [user]@[host]: Đăng nhập dưới tên người dùng được chỉ định vào máy chủ
   - -X: Bật chuyển tiếp SSH X Window System
   - -x: Tắt chuyển tiếp SSH X Window System
2. /etc/ssh/sshd_config: Cấu hình chính cho sshd
3. /etc/ssh/ssh_host_[rsa/dsa]_key: Private keys mã hóa cụ thể với quyền 600 để hạn chế truy cập cho người dùng root
4. /etc/ssh/ssh_host_[rsa/dsa]_key.pub: public keys mã hóa cụ thể với quyền 644 để hạn chế truy cập cho người dùng root và cấp quyền đọc cho người dùng khác
5. /etc/ssh/known_hosts
   - file được sử dụng để kiểm tra public key của các máy chủ đã biết/đáng tin cậy (không tồn tại theo mặc định)
6. ~/.ssh/known_hosts
   - file chứa public key của các máy chủ đã biết/đáng tin cậy đã kết nối tới/từ máy chủ hiện tại bởi người dùng sở hữu thư mục đó
7. ~/.ssh/authorized_keys
   - Lưu trữ public keys để đăng nhập dưới dạng người dùng sở hữu thư mục
8. ssh-keygen
   - Tạo cặp khóa public/private để sử dụng với SSH
   - -b [#]: Kích cỡ key mã hóa (ví dụ: 1024, 2048, v.v.)
     - -t [type]: Loại key mã hóa (DSA hoặc RSA - RSA an toàn hơn và hiện tại là mặc định)

- Sẽ yêu cầu mật khẩu - để trống sẽ cho phép bạn sử dụng khóa để đăng nhập hoàn toàn mà không cần mật khẩu, trong khi nhập một passphrase sẽ tạo ra xác thực hai yếu tố (khóa + passphrase)
- Quyền của file trên khóa nên là 644 (cũ) hoặc 600 (mới)

9. ssh-copy-id
    - Copy public key của bạn tới người dùng và máy chủ được chỉ định
        - Ví dụ: ssh-copy-id user@user.mylabserver.com
            - Sau khi nhập mật khẩu cho người dùng đã chỉ định, nó sẽ sao chép public key từ máy chủ và người dùng này vào file authorized_keys của máy chủ từ xa và người dùng đó, và sau đó bạn có thể đăng nhập vào hệ thống và tài khoản đó bằng key này
    - Phương pháp thủ công: Sao chép/dán nội dung public key của bạn vào file authorized_keys của người dùng từ xa và đặt quyền truy cập là 600
    - Kết nối tiếp theo sẽ hoạt động (không cần đặt passphrase) hoặc yêu cầu chỉ cần passphrase
10. ssh-agent
    - Bao bọc cho SSH cho phép bạn chuyển các mục (key) vào trong SSH shell để kết nối
    - Thực hiện ssh-agent để khởi động agent trên shell được chỉ định
        - **Ví dụ**: ssh-agent bash
            - Khởi động một dấu nhắc bash với agent bao bọc nó
11. ssh-add
    - Sẽ yêu cầu bạn cung cấp passphrase cho public key của bạn (nếu được thiết lập)
    - Sau khi nhập, việc sử dụng tiếp theo sẽ không đòi hỏi nhập lại cho đến khi thoát