PHỤC VỤ MỤC ĐÍCH GIÁO DỤC
FOR EDUCATIONAL PURPOSE ONLY

Làm quen với Linux

A. THỰC HÀNH

1. Thiết lập Môi trường và Làm quen Kali Linux

a. Hệ thống tập tin Kali linux

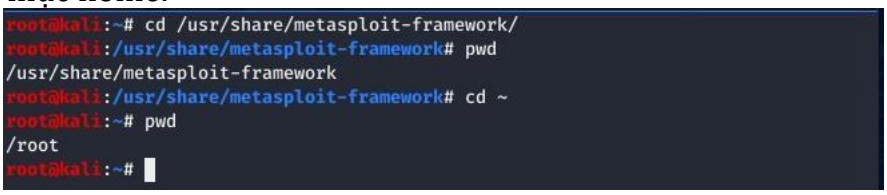
Kali Linux tuân thủ tiêu chuẩn phân cấp hệ thống tập tin (filesystem hierarchy standard – FHS) nhằm cung cấp một bố cục quen thuộc cho tất cả người dùng Linux. Các thư mục hữu ích là:

- **/bin** – các chương trình cơ bản (ls, cd, cat, …)
- **/sbin** – các chương trình hệ thống (fdisk, mkfs, sysctl, …)
- **/etc** – các tập tin cấu hình
- **/tmp** – các tập tin tạm (thường sẽ được xóa sau khi khởi động lại máy)
- **/usr/bin** – các ứng dụng (apt, ncat, nmap, …)
- **/usr/share** – hỗ trợ ứng dụng và các tập tin dữ liệu

Lệnh **ls** được sử dụng để in ra màn hình danh sách các tập tin/thư mục. Chúng ta có thể thay đổi kết quả xuất ra màn hình với hiển thị khác nhau. Tùy chọn **-a** được sử dụng để hiển thị tất cả tập tin (bao gồm tập tin ẩn) và tùy chọn **-1** hiển thị mỗi tập tin trên mỗi dòng khác nhau

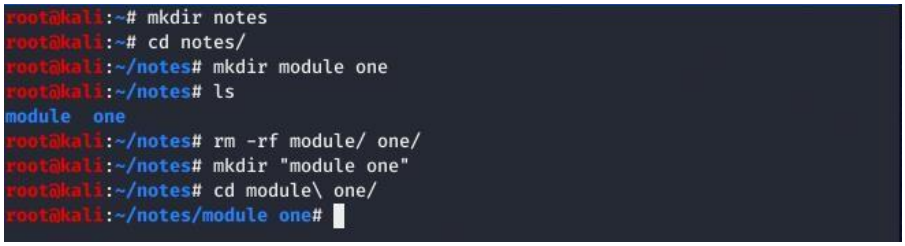
Hình 1 Liệt kê các tập tin

Linux không sử dụng các ký tự ổ đĩa như trên hệ điều hành Windows (C:\, D:\, …). Thay vào đó, tất cả các tập tin, thư mục và thiết bị đều là con của thư mục root, đại diện bởi ký tự **"/"**. Chúng ta có thể sử dụng lệnh **cd** cùng với đường dẫn để thay đổi đến thư mục được chỉ định. Lệnh **pwd** sẽ hiển thị thư mục hiện tại và chạy lệnh **cd ~** sẽ trở về thư mục home.



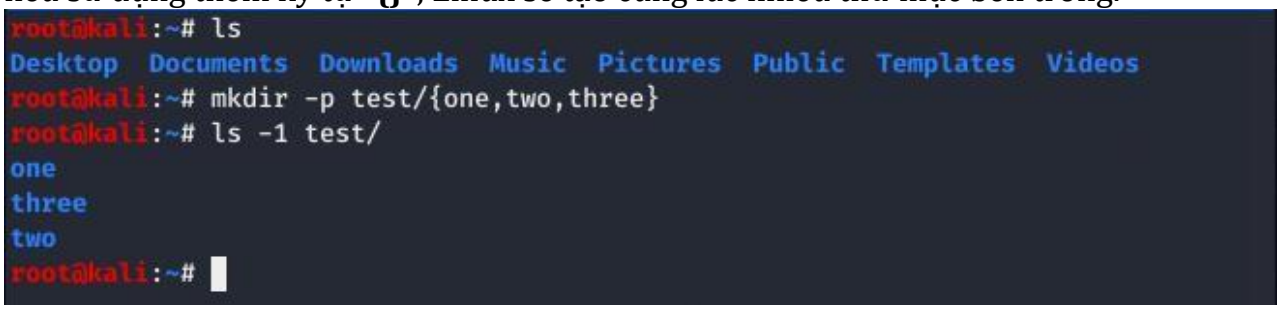
Hình 2 Di chuyển xung quanh file system

Lệnh **mkdir** đi theo sau là tên thư mục sẽ tạo ra thư mục được chỉ định. Tên của thư mục có thể chứa khoảng trắng, nhưng nên hạn chế sử dụng, thay vào đó hãy sử dụng dấu gạch nối **"-"** hoặc gạch chân **"_"**



Hình 3 Tạo thư mục trên Linux system

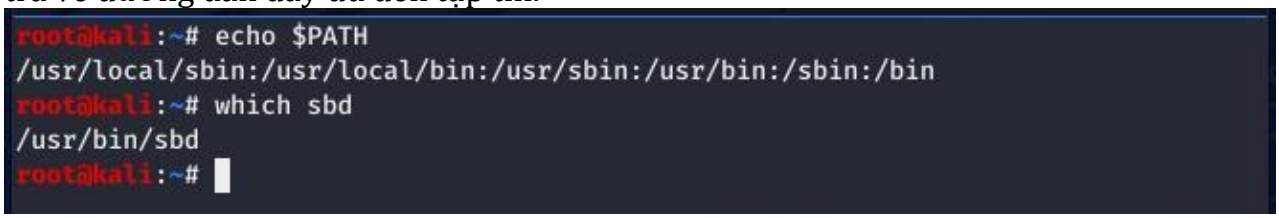
Chúng ta có thể tạo nhiều thư mục cùng 1 lúc bằng cách sử dụng tuỳ chọn **-p**, ngoài ra nếu sử dụng thêm ký tự **"{}"**, Linux sẽ tạo cùng lúc nhiều thư mục bên trong.



Hình 4 Tạo cấu trúc thư mục

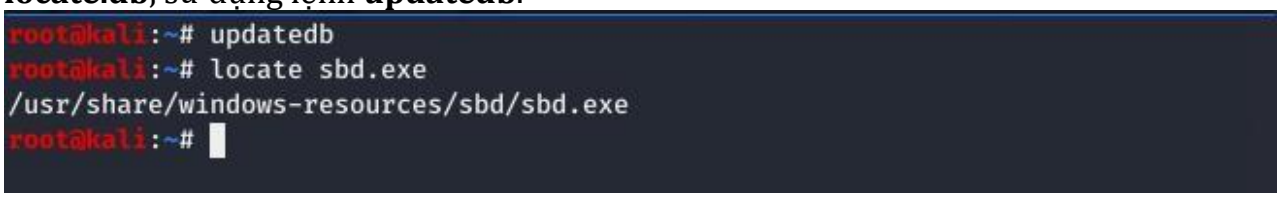
b. Tìm kiếm tập tin trong Kali linux

Lệnh **which** tìm kiếm tập tin trong các thư mục được định nghĩa trong biến môi trường $PATH. Biến này chứa danh sách các thư mục mà Kali sẽ tìm kiếm khi lệnh được đưa ra mà không chứa đường dẫn của nó. Nếu tìm thấy kết quả phù hợp, nó sẽ trả về đường dẫn đầy đủ đến tập tin.



Hình 5 Tìm kiếm tập tin sử dụng which

Cách nhanh nhất để tìm vị trí của tập tin và thư mục trong Kali là sử dụng lệnh **locate**. Để có được thời gian tìm kiếm ngắn hơn các lệnh khác, lệnh **locate** sẽ tìm kiếm trong CSDL có tên **locate.db** thay vì tìm kiếm trên toàn bộ ổ đĩa. CSDL này được tự động cập nhật thường xuyên thông qua trình lập lịch cron. Để cập nhật thủ công CSDL **locate.db**, sử dụng lệnh **updatedb**.



Hình 6 Tìm kiếm tập tin sử dụng locate

Lệnh **find** là lệnh phức tạp và linh hoạt nhất trong ba lệnh này. Việc nắm vững cú pháp của nó đôi khi có thể hơi phức tạp, nhưng khả năng của nó vượt xa các cách tìm kiếm tập tin thông thường khác. Ưu điểm của **find** so với **locate** là có thể tìm kiếm

nhiều thuộc tính hơn (kích thước, timestamp, tuổi thọ tập tin, chủ sở hữu, quyền, ..) thay vì chỉ có tên tập tin/thư mục.

```
root@kali:~# find / -name sbd*
/usr/share/windows-resources/sbd
/usr/share/windows-resources/sbd/sbd.exe
/usr/share/windows-resources/sbd/sbdbg.exe
/usr/share/doc/sbd
/usr/bin/sbd
/var/lib/dpkg/info/sbd.list
/var/lib/dpkg/info/sbd.md5sums
root@kali:~#
```

Hình 7 Tìm kiếm tập tin sử dụng lệnh find

1. Sử dụng lệnh which để xác định ví trị lưu trữ của lệnh pwd.
2. Sử dụng lệnh locate để xác định ví trí lưu trữ wce32.exe
3. Sử dụng lệnh find để xác định bất kỳ tập tin (không phải thư mục) đã được sửa đổi vào ngày trước đó.

c. Quản lý các dịch vụ

Kali Linux là một bản phân phối Linux chuyên biệt hướng đến các chuyên gia bảo mật. Như vậy, nó chứa một số tính năng không tiêu chuẩn. Cài đặt Kali mặc định đi kèm với một số dịch vụ cài đặt sẵn, chẳng hạn như SSH, HTTP, MySQL, … Do đó, các dịch vụ này sẽ được chạy tại thời điểm khởi động máy, điều này sẽ dẫn đến việc Kali hiển thị một số cổng đang mở theo mặc định mà chúng ta cần lưu ý vì các lý do bảo mật. Kali giải quyết vấn đề này bằng cách cập nhật cài đặt của nó để ngăn các dịch vụ mạng chạy cùng thời điểm khởi động máy.

Dịch vụ Secure SHell (SSH) được sử dụng phổ biến nhất để truy cập từ xa vào máy tính, sử dụng giao thức bảo mật, được mã hóa. Dịch vụ SSH dựa trên TCP và lắng nghe mặc định trên cổng 22. Để khởi động dịch vụ SSH trong Kali, chạy lệnh **systemctl** theo sau là tên dịch vụ

```
root@kali:~# sudo systemctl start ssh
root@kali:~#
```

Hình 8 Sử dụng lệnh systemctl để khởi động dịch vụ ssh

Khi lệnh được thực thi thành công, nó sẽ không trả về kết quả, nhưng chúng ta có thể kiểm chứng dịch vụ SSH đang chạy và lắng nghe trên TCP port 22 bằng cách sử dụng lệnh **ss** và chuyển tiếp kết quả sử dụng pipeline vào lệnh **grep** để tìm kiếm chữ “sshd”.

```
root@kali:~# sudo ss -anltp | grep sshd
LISTEN 0      128          0.0.0.0:22        0.0.0.0:*    users:(("sshd",pid=2076,fd=3))
LISTEN 0      128             [::]:22           [::]:*    users:(("sshd",pid=2076,fd=4))
root@kali:~#
```

Hình 9 Sử dụng lệnh ss và grep để xác nhận dịch vụ ssh đang chạy

Nếu muốn dịch vụ SSH được khởi động cùng với hệ điều hành, chúng ta sẽ kích hoạt dịch vụ sử dụng lệnh **systemctl**. Tuy nhiên, đảm bảo rằng mật khẩu mặc định trên kali đã được thay đổi.

```
root@kali:~# sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
root@kali:~#
```

Hình 10 Khởi động ssh cùng hệ điều hành sử dụng lệnh systemctl

Dịch vụ Apache HTTP thường được sử dụng trong suốt quá trình kiểm thử xâm nhập, hoặc triển khai 1 trang web, hoặc cung cấp một nền tảng để tải các tập tin lên máy của nạn nhân. Dịch vụ HTTP chạy trên TCP port 80, Để khởi động dịch vụ HTTP, sử dụng lệnh **systemctl**.

```
root@kali:~# sudo service apache2 start
root@kali:~#
```

Hình 11 Sử dụng lệnh systemctl để khởi động dịch vụ apache

Giống với dịch vụ SSH, để kiểm tra dịch vụ HTTP đang chạy và lắng nghe trên TCP port 80, sử dụng lệnh **ss** và **grep**.

```
root@kali:~# sudo ss -anltp | grep apache2
LISTEN 0      511                *:80              *:*    users:(("apache2",pid=2225,fd=4),("apache2",pid=
2224,fd=4),("apache2",pid=2223,fd=4),("apache2",pid=2222,fd=4),("apache2",pid=2221,fd=4),("apache2",pid=22
20,fd=4),("apache2",pid=2219,fd=4))
```

Hình 12 Sử dụng lệnh ss và grep để xác nhận dịch vụ apache đang chạy

Để dịch vụ HTTP khởi động cùng với hệ điều hành, sử dụng lệnh **systemctl** cùng với tùy chọn **enable**

```
root@kali:~# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.
service.
root@kali:~#
```

Hình 13 Khởi động apache cùng với hệ điều hành sử dụng lệnh systemctl

Hầu hết các dịch vụ trên Kali Linux đều hoạt động giống với với SSH và HTTP, thông qua script khởi động hoặc dịch vụ của chính nó. Để liệt kê danh sách các dịch vụ có sẵn, sử dụng lệnh **systemctl** với tùy chọn **list-unit-files**

```
root@kali:~# systemctl list-unit-files
UNIT FILE                                  STATE          VENDOR PRESET
proc-sys-fs-binfmt_misc.automount          static         enabled
-.mount                                    generated      enabled
dev-hugepages.mount                        static         enabled
dev-mqueue.mount                           static         enabled
media-cdrom0.mount                         generated      enabled
proc-sys-fs-binfmt_misc.mount              disabled       enabled
run-vmblock\x2dfuse.mount                  disabled       enabled
sys-fs-fuse-connections.mount              static         enabled
sys-kernel-config.mount                    static         enabled
sys-kernel-debug.mount                     static         enabled
sys-kernel-tracing.mount                   static         enabled
systemd-ask-password-console.path          static         enabled
systemd-ask-password-plymouth.path         static         enabled
systemd-ask-password-wall.path             static         enabled
session-2.scope                            transient      enabled
session-c1.scope                           transient      enabled
accounts-daemon.service                    enabled        enabled
anacron.service                            enabled        enabled
apache-htcacheclean.service                disabled       disabled
apache-htcacheclean@.service               disabled       disabled
```

Hình 14 Liệt kê danh sách các dịch vụ có sẵn

4. Liệt kê các port đang được mở trên Kali Linux

d. Bash Environment

Khi mở một terminal, một tiến trình Bash mới, với các biến môi trường riêng, được khởi tạo. Các biến này là một dạng lưu trữ toàn cục cho các cài đặt khác nhau được kế thừa bởi bất kỳ ứng dụng nào được chạy trong suốt phiên làm việc của terminal đó. Một trong những biến môi trường được tham chiếu phổ biến nhất là *PATH*, là danh sách các đường dẫn thư mục được phân tách bằng dấu ":" mà Bash sẽ tìm kiếm bất cứ khi nào một lệnh được chạy mà không có đường dẫn đầy đủ.
Chúng ta có thể xem nội dung của biến môi trường bằng cách sử dụng lệnh **echo** theo sau bởi ký tự **"$"** và tên biến môi trường.

```
root@kali:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
root@kali:~#
```

Hình 15 Sử dụng lệnh echo để hiển thị nội dung của biến môi trường PATH

Một số biến môi trường thông dụng như *USER, PWD,* và *HOME*, chứa giá trị lần lượt của tên user, thư mục làm việc hiện tại, và thư mục home.

```
root@kali:~# echo $USER
root
root@kali:~# echo $PWD
/root
root@kali:~# echo $HOME
/root
root@kali:~#
```

Hình 16 Sử dụng echo để liệt kê các biến môi trường USER, PWD, HOME

Biến môi trường có thể được định nghĩa sử dụng lệnh **export**. Ví dụ, nếu chúng ta tiến hành quét một đối tượng và không muốn gõ lại tên miền, chúng ta có thể gán tên miền thành biến môi trường.

```
root@kali:~# export b=google.com
root@kali:~# ping -c 2 $b
PING google.com (216.58.200.14) 56(84) bytes of data.
64 bytes from hkg12s11-in-f14.1e100.net (216.58.200.14): icmp_seq=1 ttl=114 time
=29.7 ms
64 bytes from hkg12s11-in-f14.1e100.net (216.58.200.14): icmp_seq=2 ttl=115 time
=28.3 ms

--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 28.337/29.013/29.689/0.676 ms
root@kali:~#
```

Hình 17 Sử dụng lệnh export để khai báo biến môi trường

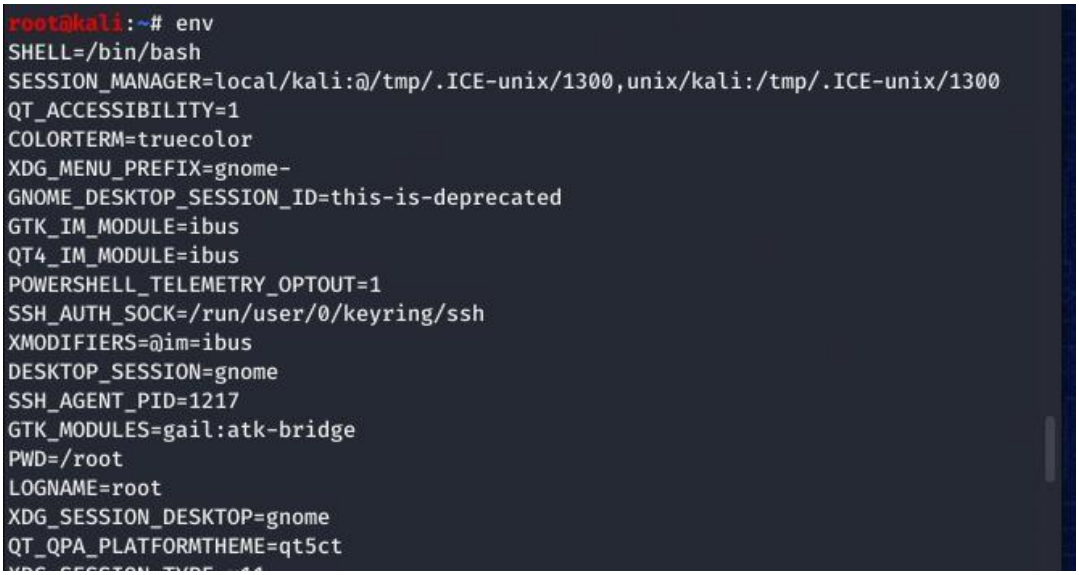
Sử dụng biến **"$$"** để hiển thị process ID của shell hiện tại nhằm đảm bảo chúng ta thực thi lệnh ở 2 shell khác nhau

```
root@kali:~# echo $$
2686
root@kali:~# var="AHIHI"
root@kali:~# echo $var
AHIHI
root@kali:~# bash
root@kali:~# echo $$
2743
root@kali:~# echo $var

root@kali:~# exit
exit
root@kali:~#
```

Hình 18 Sử dụng lệnh export để khai báo biến môi trường

Có nhiều biến môi trường được khai báo mặc định trong Kali Linux. Sử dụng lệnh **env** để xem các biến môi trường này.

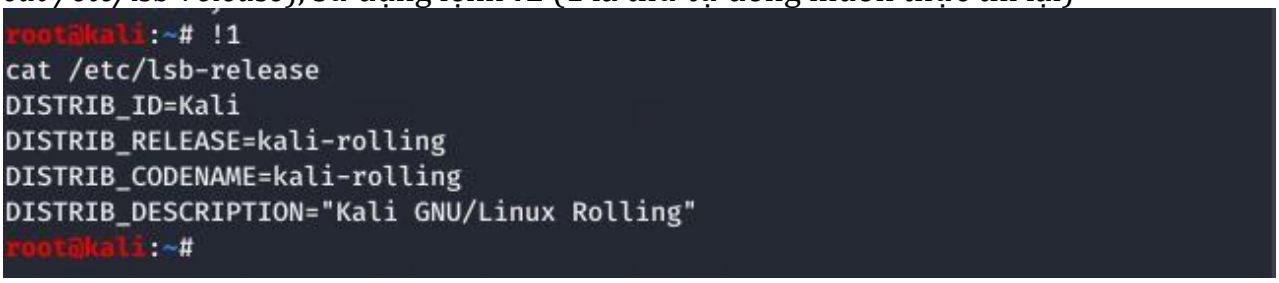


Hình 19 Sử dụng lệnh env để hiển thị tất cả biến môi trường

Bash có lưu lại lịch sử của các lệnh đã được nhập, sử dụng lệnh **history.**

Hình 20 Lệnh history

Thay vì phải gõ lại lệnh được hiển thị sau khi thực hiện lệnh **history**, chúng ta có thể sử dụng tiện ích *history expansion*. Theo Hình 21, để thực hiện lại lệnh đầu tiên (tức *cat /etc/lsb-release*), sử dụng lệnh **!1** (1 là thứ tự dòng muốn thực thi lại)



Hình 21 Sử dụng tiện ích history expansion

Ngoài ra, sử dụng lệnh **!!** để thực hiện lại lệnh trước đó (trong cùng terminal session).

Hình 22 Lập lại lệnh trước đó một cách dễ dàng

5. *Lịch sử các lệnh thực ra được lưu trữ ở đâu?*

6. *Có cách nào để ngăn chặn việc lưu trữ lịch sử lệnh hay không? Ví dụ*

e. Piping và Chuyển hướng

Mỗi chương trình chạy từ dòng lệnh (command line) đều có 3 luồng dữ liệu (data streams) được kết nối với nó, đóng vai trò là các kênh giao tiếp với môi trường bên ngoài. Các luồng này được định nghĩa theo bảng bên dưới.

Tên luồng	Mô tả
Standard Input (STDIN)	Dữ liệu được cung cấp cho chương trình
Standard Output (STDOUT)	Kết quả từ chương trình (mặc định được xuất ra terminal)
Standard Error (STDERR)	Các thông điệp lỗi (mặc định được xuất ra terminal)

Piping (sử dụng toán tử **"|"**) và chuyển hướng (sử dụng các toán tử **"<"** và **">"**) kết nối các luồng giữa các chương trình và tập tin.

Chuyển hướng đến các tập tin mới

Trong các ví dụ trước, kết quả được in ra màn hình. Trong hầu hết trường hợp, điều này rất có ích nhằm kiểm tra xem chương trình đã thực thi tới đâu. Tuy nhiên, chúng ta có thể sử dụng toán tử **">"** để lưu kết quả vào tập tin để sử dụng trong tương lai.

```
root@kali:~# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
root@kali:~# echo "AHIHI"
AHIHI
root@kali:~# echo "AHIHI" > redirection.txt
root@kali:~# ls
Desktop    Downloads  Pictures  redirection.txt  Videos
Documents  Music      Public    Templates
root@kali:~# cat redirection.txt
AHIHI
root@kali:~# echo "HELLO WORLD" > redirection.txt
root@kali:~# cat redirection.txt
HELLO WORLD
root@kali:~#
```

Hình 23 Chuyển hướng kết quả vào tập tin

Như trong Hình 24, nếu chúng ta chuyển hướng kết quả vào một tập tin không tồn tại, tập tin sẽ tự động được tạo ra. Tuy nhiên, nếu chúng ta lưu kết quả và một tập tin đã tồn tại, nội dung trong tập tin sẽ bị thay thế bằng nội dung mới.

Chuyển hướng đến tập tin đã tồn tại

Để thêm dữ liệu vào tập tin đã tồn tại (trái ngược với việc ghi đè lên tập tin), sử dụng toán tử **">>"**

```
root@kali:~# echo "THIS IS ME" >> redirection.txt
root@kali:~# cat redirection.txt
HELLO WORLD
THIS IS ME
root@kali:~#
```

Hình 24 Chuyển hướng kết quả vào tập tin đã tồn tại

Chúng ta có thể sử dụng toán tử **"<"** để gửi dữ liệu theo cách ngược lại. Trong ví dụ bên dưới, chúng ta sẽ cung cấp tham số vào lệnh **wc** bằng tập tin đã tạo trước đó. Sử dụng lệnh **wc -m** để đếm số lượng ký tự trong tập tin.

```
root@kali:~# wc -m < redirection.txt
23
root@kali:~#
```

Hình 25 Cung cấp tham số cho lệnh wc bằng toán tử <

Chuyển hướng STDERR

Theo như đặc tả kỹ thuật POSIX, các bộ mô tả tập tin (file descriptors) cho STDIN, STDOUT, STDERR được định nghĩa là 0, 1 và 2. Những con số này rất quan trọng vì chúng có thể được sử dụng để kiểm soát các luồng dữ liệu tương ứng từ dòng lệnh trong khi thực thi hoặc nối các lệnh khác nhau với nhau. Để hiểu rõ hơn về cách hoạt động của các con số của bộ mô tả tập tin, hãy xem xét ví dụ chuyển hướng lỗi chuẩn (STDERR) như sau:

```
root@kali:~# ls .
Desktop    Downloads  Pictures  redirection.txt  Videos
Documents  Music      Public    Templates
root@kali:~# ls -al test/
ls: cannot access 'test/': No such file or directory
root@kali:~# ls -al test/ 2> error.txt
root@kali:~# cat error.txt
ls: cannot access 'test/': No such file or directory
root@kali:~# mkdir -p test/{one,two,three}
root@kali:~# ls -a1 test/ 2>> error.txt
.
..
one
three
two
root@kali:~# cat error.txt
ls: cannot access 'test/': No such file or directory
root@kali:~#
```

Hình 26 Chuyển hướng STDERR vào tập tin

Theo Hình 27, tập tin **error.txt** chỉ chứa các thông điệp lỗi (được tạo ra trên *STDERR*) bằng cách thêm vào số 2 trước toán tử ">" (*2=STDERR*)

Piping

Tiếp tục với ví dụ sử dụng lệnh **wc**, chúng ta hãy xem cách chuyển hướng kết quả từ lệnh trước thành tham số đầu vào cho lệnh kế tiếp. Hãy quan sát ví dụ bên dưới:

```
root@kali:~# cat error.txt
ls: cannot access 'test/': No such file or directory
root@kali:~# wc -m < error.txt
53
root@kali:~# cat error.txt | wc -m
53
root@kali:~# cat error.txt | wc -m > output.txt
root@kali:~# cat output.txt
53
root@kali:~#
```

Hình 27 Piping kết quả của lệnh cat vào trong lệnh wc

Trong Hình 28, chúng ta sử dụng ký tự pipe **"|"** để chuyển hướng kết quả của lệnh **cat** thành tham số đầu vào của lệnh **wc**. Khái niệm này có vẻ tầm thường nhưng kết hợp các lệnh khác nhau lại với nhau lại là một công cụ mạnh mẽ để kiểm soát tất cả loại dữ liệu

> 7. *Sử dụng lệnh cat cùng với lệnh sort để sắp xếp lại nội dung của tập tin /etc/passwd, sau đó lưu kết quả vào một tập tin mới có tên passwd_new và thực hiện đến số lượng dòng có trong tập tin mới.*

f. Tìm kiếm và thao tác văn bản

Lệnh **grep** thực hiện tìm kiếm các tập tin văn bản để tìm sự xuất hiện của một biểu thức chính quy (regular expression) cung cấp trước và xuất ra kết quả tương ứng.

Một số tùy chọn phổ biến bao gồm **-r** để tìm trong các thư mục con, và **-i** để bỏ qua kiểu chữ (hoa, thường).

```
root@kali:~# ls -la /usr/bin | grep zip
-rwxr-xr-x  1 root root          39784 Dec 28  2019 fcrackzip
-rwxr-xr-x  1 root root          14600 Dec 28  2019 fcrackzipinfo
-rwxr-xr-x  1 root root          22792 Jul 27  2019 funzip
-rwxr-xr-x  1 root root           3516 Mar 23 15:05 gpg-zip
-rwxr-xr-x  1 root root           4754 Aug  9  2019 p7zip
-rwxr-xr-x  1 root root           5656 Oct 22  2019 preunzip
-rwxr-xr-x  1 root root           5656 Oct 22  2019 prezip
-rwxr-xr-x  1 root root          14488 Oct 22  2019 prezip-bin
-rwxr-xr-x  2 root root         183136 Jul 27  2019 unzip
-rwxr-xr-x  1 root root          84664 Jul 27  2019 unzipsfx
-rwxr-xr-x  1 root root         213136 Aug 16  2015 zip
-rwxr-xr-x  1 root root          90432 Aug 16  2015 zipcloak
-rwxr-xr-x  1 root root          50718 Jun  7 03:56 zipdetails
-rwxr-xr-x  1 root root           2953 Jul 27  2019 zipgrep
-rwxr-xr-x  2 root root         183136 Jul 27  2019 zipinfo
-rwxr-xr-x  1 root root          86048 Aug 16  2015 zipnote
-rwxr-xr-x  1 root root          86048 Aug 16  2015 zipsplit
root@kali:~#
```

Hình 28 Tìm kiếm bất kỳ tập tin nào trong /usr/bin có chứa chữ "zip"

Lệnh **sed** là một trình chỉnh sửa luồng mạnh mẽ. Ở cấp độ cao, lệnh **sed** thực hiện chỉnh sửa văn bản trên một luồng văn bản, hoặc một tập hợp các tập tin được chỉ định hoặc ở STDOUT.

```
root@kali:~# echo "Hello world" | sed 's/world/Vietnam/'
Hello Vietnam
root@kali:~#
```

Hình 29 Thay thế từ trong output stream sử dụng lệnh sed

Lệnh **cut** được sử dụng để trích xuất một phần văn bản từ 1 dòng và xuất nó ra STDOUT. Một số thuộc tính được sử dụng phổ biến bao gồm **-f** cho thứ tự trường muốn lấy và **-d** cho ký tự muốn phân cách.

```
root@kali:~# echo "I love maths,physics,chemistry and literature" | cut -d "," -f 2
physics
root@kali:~#
```

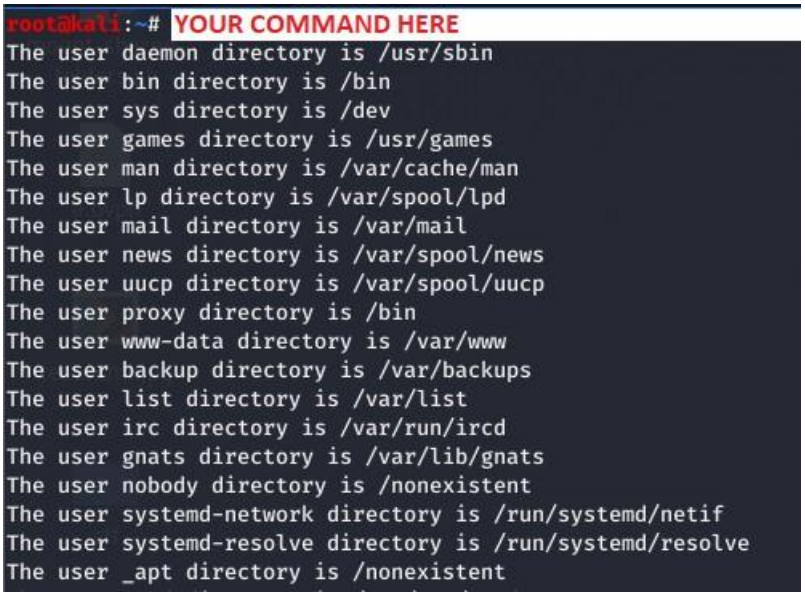
Hình 30 Trích xuất các trường từ lệnh echo sử dụng lệnh cut

AWK là ngôn ngữ lập trình được thiết kế để xử lý văn bản và thường được sử dụng làm công cụ báo cáo và trích xuất dữ liệu. Nó cũng cực kỳ mạnh mẽ và khá phức tạp. Một tùy chọn thường được sử dụng với lệnh awk là **-F**, là dấu phân cách giữa các trường, và lệnh **print**, xuất kết quả ra STDOUT.

```
root@kali:~# echo "hetto::there::friend" | awk -F "::" '{print $1, $3}'
hetto friend
root@kali:~#
```

Hình 31 Trích xuất các trường từ stream sử dụng lệnh awk

8. *Sử dụng tập tin /etc/passwd, trích xuất tên user và home directory cho tất cả user có shell được thiết lập là /usr/sbin/nologin. Lưu ý, chỉ sử dụng 1 dòng lệnh duy nhất. Kết quả xuất ra màn hình như hình dưới.*



Hình 32 Các thư mục home của user với shell là /usr/sbin/nologin

9. *Tải tập tin access_log.txt.gz tại*

(https://github.com/blakduk/ahihi/raw/master/access_log.txt.gz), sau đó thực hiện liệt kê danh sách các địa chỉ IP và số lượng tương ứng, thực hiện sắp xếp giảm dần

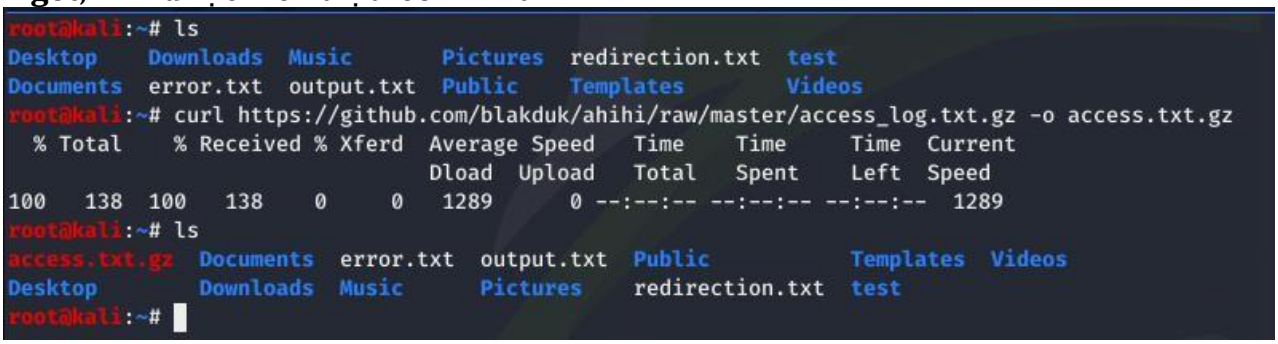
Hình 33 Liệt kê danh sách địa chỉ IP cùng số lượng tương ứng

g. Tải tập tin

Lệnh **wget** được sử dụng thường xuyên để tải các tập tin sử dụng giao thức HTTP/HTTPS và FTP. Sử dụng tùy chọn **-O** để lưu kết quả vào tập tin với tên khác

Hình 34 Tải xuống tập tin sử dụng lệnh wget

Curl là một công cụ dùng để truyền dữ liệu đến hoặc từ máy chủ sử dụng một loạt các giao thức bao gồm IMAP/S, POP3/S, SCP, SFTP, SMB/S, SMTP/S, TELNET, TFTP và các giao thức khác. Pentester có thể sử dụng công cụ này để tải xuống hoặc tải lên các tập tin và tạo ra các request phức tạp. Các sử dụng cơ bản nhất của nó cũng giống với **wget**, như được hiển thị theo hình dưới.



Hình 35 Tải xuống tập tin sử dụng lệnh curl

10. Hãy cho biết đường dẫn thực thi của 2 lệnh wget và curl?

11. Có thể sử dụng lệnh curl để thay đổi các HTTP header được hay không? Nếu được, cho ví dụ?